《붉은별》 봉사기용체계 3.0
보안방책 관리지도서

차 례

머리말

위대한 령도자 **김정일**동지께서는 다음과 같이 지적하시였습니다.

《프로그람을 개발하는데서 기본은 우리 식의 프로그람을 개발하는것입니다. 우리는 우리 식의 프로그람을 개발하는 방향으로 나가야 합니다.》

(《**김정일**선집》 제15권, 196페지)

지금 우리 나라에서는 지식경제시대의 요구에 맞게 정보산업을 자립적으로 건설하고 발전시키기 위한 투쟁이 광범히 벌어지고있으며 이 과정에 핵심기술의 집합체인 우리 식 조작체계개발사업에서 커다란 성과들이 이룩되고있습니다. 이러한 우리 식의 조작체계에 우리 식의 보안을 실현하는것은 과학기술발전에서 주체를 세우고 나라의 방위력을 강화하기 위하여 절실하게 제기되는 매우 중요한 문제입니다.

《붉은별》봉사기용체계 3.0 판에서는 보안조작체계를 리용하여 우리 식의 보안방책을 작성하여 실현함으로써 기밀성과 완전성을 우리 식으로 확고히 보장하고있습니다.

이 사용지도서는 《붉은별》봉사기용체계 3.0 판을 처음으로 사용하는 보안관리자들이 보안방책을 자체의 실정에 맞게 작성하여 사용할수 있도록 하기 위하여 작성한 지도서입니다. 이 지도서에서는 보안조작체계개념과 보안방책지령들의 기초로부터 시작하여 간단한 실례들을 줌으로써 보안관리자들이 자체로 해당 기관의 특성에 맞게 보안방책을 작성하거나 수정할수 있도록 준비하는데 도움을 줍니다.

제 1 장에서는 보안조작체계에 대한 개념을 서술합니다.

제 2 장에서는 보안방책의 원리들을 보안조작체계의 문맥과 결부하여 서술합니다.

제3장에서는 보안조작체계에서 리용하는 보안방법들의 류형을 2가지로 분류하고 설명합니다.

제 4 장에서는 보안조작체계의 구성과 관리에 대하여 구체적으로 서술합니다.

제 5, 6 장에서는 자체로 방책작성을 할수 있는 기초를 주기 위하여 보안방책들의 간단한 실례를 들어 해설합니다.

제 7 장에서는 보안조작체계사용에서 제기되는 일련의 문제들을 취급하였습니다.

제 1 장 소개

보안조작체계(SELinux)는 봉사기핵심부에서 강제접근조종(MAC)을 실현한것입니다. 보안조작체계를 리용하면 봉사기체계내에서 화일들과 프로쎄스들, 그의 조작들에 대한 규칙들을 정의한 방책에 의하여 보안을 강화할수 있습니다.

보안조작체계를 사용할 때 등록부와 장치들을 포함한 모든 화일들은 다 객체로 간주되며 사용자가 실행하는 지령이나 웨브열람기와 같은 프로쎄스들은 다 주동체로 간주됩니다. 대부분의 조작체계들은 주동체가 객체들과 호상작용하는 방법과 주동체들의 호상작용방법을 조종하는 자유접근조종(DAC)체계를 사용합니다. 자유접근조종을 사용하는 조작체계들에서 사용자들은 소유자가 자기의것으로 되여있는 화일들(객체들)에 대한 허가권한을 조종합니다. 즉 사용자들은 자기의 사용자등록부를 모두가 열람도록할수 있는데 이 경우 사용자들과 프로쎄스들(주동체들)은 중요한 정보들에 접근할수 있으며 결국 사용자가 요구하지 않은 조작들에 대한 방지가 전혀 진행되지 않습니다.

강력한 체계보안을 위해서는 자유접근조종 하나만에 의존하는것이 좋지 않다. 자유접근조종접근판정들은 사용자신분정보와 소유권에만 기초하며 다른 보안관련정보인 사용자의 역할, 프로그람의 기능과 신뢰성, 자료의 민감성과 완전성 등은 무시합니다. 매 사용자는 표준적으로 자기의 화일들에 대해서는 조작의 완전한 자유를 가지는데 이로 하여 체계전반적인 보안강화가 어려워집니다. 더우기 사용자에 의하여 실행되는 모든 프로그람들은 그 사용자에게 부여된 모든 권한들을 계승하여 사용자의 화일들에 대한 접근을 자유로 변경시킬수 있습니다.

다음 실례는 보안조작체계를 실행하지 않는 봉사기조작체계에서 사용되는 허가권한들의 실례입니다. ls –l지령을 사용하면 화일의 허가권한들을 볼수 있습니다.

```
$ ls -l file1
-rwxrw-r-- 1 user1 group1 0 2012-08-30 11:03 file1
```

이 실례에서 첫 세개의 허가권한비트들인 rwx는 봉사기의 user1 사용자(이 경우에는 소유자)가 file1 에 대하여 가지는 접근을 조종합니다. 다음의 세개의 허가권한비트들인 rw-는 봉사기의 group1 집단이 file1 에 대하여 가지는 접근을 조종합니다. 마지막 세개의 허가권한비트들인 r--는 모든 그밖의 사용자들이 file1 에 대하여 가지는 접근을 조종하는데 이때 그밖의 사용자들에는 모든 사용자들과 프로쎄스들이 포함됩니다.

보안조작체계에서는 봉사기핵심부에 강제접근조종(MAC)기능을 추가하며 《붉은별》봉사기용체계 3.0 판에서 기정으로 시행됩니다. 일반적인 강제접근조종구성방식에서는 체계의 모든 프로쎄스들과 화일들에 대한 관리적모임의 보안방책을 강화하는 능력이 필요하며 이것은 각종 보안관련정보들을 포함하는 표식들에 기초하여 접근판정을 진행합니다. 강제접근조종이 정확히 실현되면 체계가 자체를 충분히 방어할수 있게 하며 보안관련응용프로그람들을 부당하게 변경시키거나 악성코드들이 그것을 우회하는것을 방지함으로써 응용프로그람에 대한 강력한 보안을 제공합니다. 강제접근조종은 신뢰할수 없는 응용프로그람들이 안전하게 실행되게 하는 강력한 응용프로그람분리기능을 제공합니다. 실행중의 프로쎄스들과 관련된 특권준위들을 제한하는 능력은 응용프로그람들과 체계봉사들에서의 취약점을 악용하는것으로 하여 초래될수 있는 잠재적인 손상범위를 제한합니다. 강제접근조종은 제한된 권한을 가진 합법적인 사용자로부터 정보가 보호될수 있게 하며 마찬가지로 무의식적으로 악의적인 응용프로그람들을 실행한 사용자로부터도 정보가 보호될수 있게 합니다.

다음 실례는 보안조작체계를 사용하는 봉사기조작체계들에서 프로쎄스들과 봉사기사용자들, 화일들에 사용되는 보안관련정보들을 포함하는 표식들의 한가지 실례입니다. 이 정보는 보안조작체계문맥(SELinux Context)이라고 불리우는데 이것은 ls –Z지령을 사용하면 알수 있습니다.

```
$ ls  -Z  file1
-rw-rw-r--.     kkh  kkh    user_u:object_r:user_home_t:s0          file1
```

이 실례에서 보안조작체계는 사용자(user_u), 역할(object_r), 형(user_home_t), 준위(s0)를 제공합니다. 이 정보는 접근조종을 결정하는데 사용됩니다. 자유접근조종에서는 봉사기사용자와 집단ID들에만 기초하여 접근이 조종됩니다. 중요한것은 보안조작체계규칙들이 자유접근조종규칙들 다음에 검사된다는것입니다. 보안조작체계규칙들은 자유접근조종규칙들이 접근을 거부한다면 사용되지 않습니다.

봉사기체계와 보안조작체계사용자

보안조작체계를 실행하는 봉사기조작체계들에는 봉사기사용자들이 있으며 이와 마찬가지로 보안조작체계사용자들도 존재합니다. 보안조작체계사용자들은 보안조작체계방책부분품입니다. 봉사기사용자들은 보안조작체계사용자들과 대응됩니다. 혼잡을 피하기 위하여 이 지도서에서는 봉사기사용자와 보안조작체계사용자라는 술어를 사용하여 두 개념을 구별합니다.

제 1 절 보안조작체계의 우점

모든 프로쎄스들과 화일들에는 하나의 형태를 가진 표식이 붙습니다. 프로쎄스들에 할당되는 표식은 령역(Domain)이라고 정의하고 화일들에 할당되는 표식은 형(Type)이라고 정의합니다. 프로쎄스들은 자기의 령역내에서 실행됨으로써 서로 구분되며 보안조작체계규칙들은 프로쎄스들이 어떻게 화일들과 호상작용하는가를 정의하고 마찬가지로 프로쎄스들이 어떻게

다른 프로쎄스들과 호상작용하는가를 정의합니다. 접근은 그것을 명확히 허가하는 보안조작체계규칙들이 존재하는 경우에만 허가됩니다.

보안조작체계규칙은 관리적인 관점에서 정의되고 체계전반적인것으로써 사용자가 자유롭게 설정할수 없습니다.

또한 특권준위확대공격들에 대한 취약점을 감소시킵니다. 한가지 실례로 프로쎄스들이 자기의 특정의 령역들에서 실행되기때문에 따라서 서로 분리되며 보안조작체계규칙들에서 프로쎄스들이 화일들과 다른 프로쎄스들에 어떻게 접근하는가를 정의하기때문에 만일 한 프로쎄스가 공격당하면 공격자는 그 프로쎄스의 일반기능에 대한 접근과 프로쎄스가 접근할수 있는 구성화일들에 대한 접근만을 가집니다. 실례로 만일 Apache HTTP봉사기가 공격당하면 공격자는 사용자등록부들내의 화일들을 읽는데 그 프로쎄스를 사용할수 없으며 이것은 특정한 보안조작체계규칙들이 추가되거나 그러한 접근을 허가하도록 구성하지 않았으면 불가능합니다.

보안조작체계는 자료의 기밀성과 완전성을 강화하는데 사용될수 있으며 마찬가지로 신뢰되지 않는 입력물로부터 프로쎄스들을 보호합니다.

그러나 보안조작체계는 명백히 다음과 같이 생각해서는 안됩니다.

- 반비루스쏘프트웨어
- 통과암호처리
- 방화벽 혹은 기타 다른 보안체계의 교체물
- 모든것이 하나로 묶어진 보안해결책

보안조작체계는 현존 보안해결책들의 향상을 위하여 설계되였지 완전한 대책이라고 말할수는 없습니다. 보안조작체계를 실행하고있다고 해도 전통적인 보안규정들을 계속 유지하여야 합니다. 즉 반비루스왁찐을 최신판

본으로 유지하고 예측불가능한 통과암호들을 사용하여야 하며 방화벽 등을 갖추어야 합니다.

제 2 절 보안조작체계의 간단한 실례

다음의 실례들은 보안조작체계가 어떻게 보안을 향상시키는가를 보여줍니다.

보안조작체계의 기정동작은 거부입니다. 만일 어떤 화일을 여는 프로쎄스에 대하여 본다면 접근을 허가하는 보안조작체계방책이 존재하지 않을 때에 접근은 거부됩니다.

보안조작체계는 봉사기사용자들을 제한합니다. 보안조작체계방책에는 많은 제한된 보안조작체계사용자들이 존재합니다. 실례로 봉사기사용자를 보안조작체계의 user_u사용자에로 대응시키면 sudo와 su와 같은 사용자ID설정(setuid)프로그람들을 실행할수 없는 봉사기사용자로 됩니다. 마찬가지로 사용자들이 자기의 사용자등록부에서 화일들과 응용프로그람들을 실행하지 못하게 합니다. 이와 같이 설정되는 경우에는 사용자들이 자기의 사용자등록부들로부터 악의적인 화일들을 실행하지 못하게 됩니다.

다음으로 보안조작체계는 프로쎄스분리를 실현합니다. 프로쎄스들은 자기의 령역내에서 실행되며 프로쎄스들이 다른 프로쎄스들이 사용하는 화일들에 접근하지 못하도록 합니다. 마찬가지로 프로쎄스들이 다른 프로쎄스들에 접근하지 못하게 합니다. 보안조작체계는 구성상 실수로 인한 손실을 제한해줍니다. 령역이름체계(DNS)봉사기들은 서로 정보들을 자주 반복응답하는데 그로 하여 지역이동(zone transfer)이 있게 됩니다. 공격자들은 지역이동들을 리용하여 거짓정보를 가지고 DNS봉사기들을 갱신할수 있습니다. 《붉은별》봉사기용체계 3.0 판에서 DNS봉사기와 같은 Berkeley

Internet Name Domain(BIND)을 실행할 때 관리자가 만일 어느 봉사기가 지역이동을 할수 있는가를 제한하는것을 잊어버리는 경우 기정의 보안조작체계방책은 BIND의 named대몬자체와 다른 프로쎄스들에 의한 지역이동들을 통하여 지역(zone)화일들이 갱신되는것을 막아줍니다.

제 3 절 보안조작체계의 구성방식

보안조작체계는 핵심부에 내장된 하나의 보안모듈입니다. 보안조작체계는 적재가능한 방책들에 의하여 실현됩니다. 보안관련접근이 발생할 때 즉 어떤 프로쎄스가 어떤 화일을 열려고 할 때 그 조작은 보안조작체계에 의하여 핵심부에서 차단되게 됩니다. 만일 어떤 보안조작체계의 방책규칙이 그 조작을 허가한다면 조작이 계속되게 되며 그렇지 않으면 조작이 차단되여 그 프로쎄스는 오유를 되돌려받습니다.

보안조작체계판정들 즉 접근의 허가나 거부와 같은것들은 고속완충기억(cache) 됩니다. 이 고속완충기억기를 접근벡토르고속완충기(AVC)라고 합니다. 판정결과를 완충기억하는것은 보안조작체계규칙들을 판정하는 빈도수를 감소시키며 성능을 증가시키기 위해서입니다. 보안조작체계규칙들은 자유접근조종규칙이 처음부터 접근을 거절하는 경우에는 효과가 없다는것을 잊지 말아야 합니다.

제 2 장 보안조작체계의 문맥

프로쎄스들과 화일들에는 하나의 보안조작체계문맥을 가진 표식들이 붙여지게 되는데 이 보안문맥에는 보안조작체계의 사용자, 역할, 형 그리고 준위(선택가능)와 같은 추가적인 정보들이 포함되여있습니다. 보안조작체계를 실행할 때 이 모든 정보들은 접근조종판정들을 산출하기 위하여 리용됩니다. 《붉은별》봉사기용체계 3.0 판에서 보안조작체계는 역할기초의 근조종(RBAC), 형시행(TE), 그리고 추가적으로 다중준위보안(MLS)들이 결합된 하나의 결합물을 제공합니다.

아래에 보안조작체계문맥을 보여주는 실례가 있습니다.

보안조작체계문맥들은 보안조작체계를 실행하는 봉사기조작체계들에서 프로쎄스들과 봉사기사용자들, 화일들에 대하여 적용됩니다. 화일들과 등록부들의 보안조작체계문맥을 보려면 ls –Z 지령을 사용하여야 합니다.

```
$ ls  -Z  file1
-rw-rw-r--.     kkh  kkh    user_u:object_r:user_home_t:s0            file1
```

보안조작체계문맥들은 보안조작체계의 《사용자:역할:형:준위》라는 문법에 따릅니다.

보안조작체계사용자

보안조작체계사용자신분정보는 고유한 역할모임에 대한 권한과 고유한 MLS범위에 대한 권한을 가진 방책으로 알려진 신분정보입니다. 매 봉사기사용자는 보안조작체계방책을 통하여 하나의 보안조작체계사용자에 대응됩니다. 대응된 보안조작체계사용자신분정보는 그 쎄션에서 프로쎄스들에 대한 보안조작체계문맥에 사용되며 그것은 사용자들의 역할과 그들이 활동할수 있는 준위를 정의하기 위해서입니다. 보안조작체계와 봉사기사용자들의 계산자리사이의 대응목록을 보려면 봉사기체계의 root사용자로

semanage login –l 지령을 실행시키면 된다(그림 1). 《붉은별》봉사기용체계 3.0 판에서는 이 지령을 보안관리자역할(secadm_r)을 가진 root관리자만이 실행할수 있습니다(3 장 2 절 참고). 편리상 보안관리자역할을 가진 root관리자를 보안관리자, 체계관리자역할(sysadm_r)을 가진 root관리자를 체계관리자라고 서술합니다.

```
《붉은별》봉사기용체계 3.0판
핵심부 2.6.32-120727.RSS3.i686 (i686)
localhost login: root
Password:
Last login: Wed Dec 12 11:38:21 on tty2
[root@localhost ~]# newrole -r secadm_r
암호:
[root@localhost ~]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@localhost ~]# semanage login -l

가입사용자이름        보안리눅스사용자       MLS/MCS범위

__default__          user_u                 s0
root                 root                   s0-s15:c0.c1023
system_u             system_u               s0-s15:c0.c1023
testuser             xguest_u               s0
[root@localhost ~]# _
```

그림 1. 봉사기사용자와 보안조작체계사용자사이의 대응관계

그림 1 에서 《가입사용자이름》렬은 봉사기사용자들을 목록화하며 《보안조작체계사용자》렬은 봉사기사용자에 어느 보안조작체계사용자가 대응되는가를 목록화합니다. 프로쎄스들에 대해서 보안조작체계사용자는 접근할수 있는 역할과 준위를 제한받게 됩니다. 마지막 렬인 《MLS/MCS범위》렬은 다중준위보안(MLS)과 다중종류보안(MCS)이 사용하는 준위입니다.

역할

보안조작체계의 중요부분은 역할기초의 접근조종(RBAC)모형입니다. 여기서 역할이라는것은 역할기초의 접근조종에서 리용하는 보안속성입니다. 보안조작체계사용자들은 역할들에 대한 권한을 가지고있으며 역할들은 령

역들에 대하여 권한을 가지고있습니다. 이러한 역할은 령역들과 보안조작체계사용자들사이의 내용물로서 봉사합니다. 역할들은 어느 령역에 들어갈수 있는가를 결정하며 결국 이것은 어떤 형의 객체들에 접근할수 있는가를 조종합니다. 이것은 특권준위확대공격들에 대한 취약점을 줄이는데 도움이 되는것입니다.

형

형이라는것은 형시행에서 리용하는 보안속성입니다. 형은 프로쎄스들에 대하여서는 령역을 정의하며 화일들에 대하여서는 형을 정의합니다. 보안조작체계규칙(형시행규칙)들은 형들이 서로 접근할수 있는 방법을 정의하고있습니다. 즉 그것이 형에 접근하는 령역인가 혹은 다른 령역에 접근하는 령역인가를 정의합니다. 접근은 그것을 허가하는 고유한 보안조작체계규칙이 존재하는 경우에만 허가됩니다.

준위

이 준위라는것은 다중준위보안과 다중범주(Category)보안에서 리용하는 보안속성입니다. 다중준위보안에서 리용하는 범위는 준위의 쌍으로서 준위들이 차이나면 《낮은준위-높은준위》로 쓰고 준위들이 동등하면《낮은준위》로 쓴다(s0-s0 은 s0 과 같다). 매 준위는 기밀성준위(Sensitivity)-범주(Category)의 쌍으로 이루어지는데 범주들을 추가선택적으로 가질수 있습니다. 만일 범주들이 있다면 그때 준위는《기밀성준위:범주모임》으로 작성됩니다. 범주들이 없는 경우에는《기밀성준위》로 작성됩니다.

만일 범주모임이 련속적인 계렬이라면 간략하여 표시할수 있습니다. 실례로 c0.c3 은 c0,c1,c2,c3 과 같습니다. /etc/selinux/targeted/setrans.conf화일은 준위들인 (s0:c0)을 사람이 리해하기 쉬운 형태(즉 CompanyConfidential)에 대응시킵니다. 분문편집기로 setrans.conf화일을 편집하지 말아야 하며 변경시키려면 semanage를 사용하여야 합니다. 이에 대하여서는 semanage에

대한 안내페지를 참고할수 있습니다.《붉은별》봉사기용체계 3.0 판에서 다중범주보안은 1024개의 서로 다른 범주들, 즉 c0부터 c1023까지를 지원합니다. s0-s0:c0.c1023 은 기밀성준위 s0 이며 모든 범주들에 대하여 권한을 가진다는것입니다.

다중준위보안은 Bell-LaPadula모형을 강화한것으로서 《역할기초의 접근조종에 대한 보호프로화일》(LSPP)의 환경변수들에서 사용됩니다. 다중준위보안을 리용하기 위해서는 selinux-policy-mls패키지를 설치하여야 하며 다중준위보안이 기정의 보안조작체계방책이 되도록 구성하여야 합니다.《붉은별》봉사기용체계 3.0 판에는 MLS전용방책을 제공하지 않습니다.

제 1 절 령역이행

새로운 령역에 대하여 입구점(entrypoint)형을 가지는 어떤 응용프로그람을 실행시키면 한 령역에서의 프로쩨스가 다른 령역으로 이행합니다. 입구점실행권한은 보안조작체계방책에서 어떤 령역에 들어가는데 어느 응용프로그람이 사용될수 있는가를 조종합니다. 다음의 실례는 령역이행에 대하여 설명해줍니다.

1. 사용자는 자기들의 통과암호를 변경하려고 합니다. 그러자면 passwd 응용프로그람을 실행시켜야 합니다. /usr/bin/passwd라는 실행가능화일은 passwd_exec_t형으로 표식되여있습니다.

```
# ls -Z /usr/bin/passwd
-rwsr-xr-x   root  root  system_u:object_r:passwd_exec_t:s0   /usr/bin/passwd
```

passwd응용프로그람은 /etc/shadow에 접근하는데 이 화일은 shadow_t형으로 표식되여있습니다.

```
# ls -Z /etc/shadow
----------. root  root  system_u:object_r:shadow_t:s0       /etc/shadow
```

2. 보안조작체계방책규칙에는 passwd_t령역에서 실행되는 프로쩨스들이 shadow_t형으로 표식된 화일들을 읽거나 쓰기할수 있는가 하는 허가권한이 서술됩니다. shadow_t형은 통과암호변경을 위하여 요구되는 화일들에만 적용됩니다. 여기에는 /etc/gshadow, /etc/shadow 그리고 이것들의 여벌복사화일들이 포함됩니다.

3. 보안조작체계방책규칙에는 passwd_t령역이 passwd_exec_t형에 입구점실행권한을 가진다는것이 서술됩니다.

4. 어떤 사용자가 /usr/bin/passwd응용프로그람을 실행시킬 때 사용자의 쉘프로쩨스는 passwd_t령역에로 이행합니다. 보안조작체계를 실행하고있으

면 기정동작은 거부하는것이기때문에 passwd_t령역에서 실행되는 응용프로그람들에서 shadow_t형으로 표식된 화일들에 접근할수 있게 하는 어떤 규칙이 존재하며 따라서 passwd응용프로그람은 /etc/shadow에 접근할수 있고 사용자의 통과암호를 갱신할수 있습니다.

이 실례는 단지 령역이행을 설명하기 위한 기초실례로 사용됩니다. 비록 passwd_t령역에서 실행되는 주동체들이 shadow_t화일형태로 표식된 객체들에 접근할수 있도록 허가하는 어떤 실제적인 규칙이 있다고 해도 그 주동체가 새로운 령역에로 이행할수 있기 전에 다른 보안조작체계방책규칙들이 맞물려져야 합니다. 이 실례에서 형시행은 다음과 같은것들을 담보합니다.

- passwd_t령역에는 passwd_exec_t형으로 표식된 응용프로그람을 실행하여야만 들어갈수 있고 lib_t형으로 권한이 부여된것은 공유서고들로부터만 실행될수 있으며 임의의 다른 응용프로그람들로부터는 실행할수 없습니다.
- passwd_t와 같은 권한이 부여된 령역들만이 shadow_t형으로 표식된 화일들에 쓰기할수 있습니다. 지어 다른 프로쎄스들이 상급사용자권한을 가지고 실행되고있다고 해도 그 프로쎄스들은 shadow_t형으로 표식된 화일들에 쓰기를 할수 없으므로 passwd_t령역에서 실행되지 않습니다.
- 권한이 부여된 령역들만이 passwd_t령역에 이행할수 있습니다. 실례로 sendmail_t령역에서 실행되고있는 sendmail프로쎄스는 passwd를 실행할 합법적인 리유가 없기때문에 따라서 passwd_t령역에 절대로 이행할수 없습니다.
- passwd_t령역에서 실행되고있는 프로쎄스들은 etc_t나 shadow_t형으로 표식된 화일들과 같은 권한이 부여된 형태들에 대하여서만 읽거

나 쓰기를 진행할수 있습니다. 이렇게 하면 passwd응용프로그람이 속임수로 임의의 화일을 읽거나 쓰기할수 없게 할수 있습니다.

제 2 절 프로쎄스에 대한 보안문맥

프로쎄스들에 대한 보안조작체계문맥을 보려면 ps –eZ지령을 리용하여야 합니다. 실례는 다음과 같습니다.

1. 가상말단(tty1)에서 /usr/bin/passwd지령을 실행합니다. 새로운 통과암호는 입력하지 않습니다.

2. 다른 가상말단(tty2)에서 ps –eZ | grep passwd지령을 실행합니다. 출력은 다음과 같이 나올수 있습니다.

```
root:sysadm_r:passwd_t:s0-s15:c0.c1023   4129  tty1  00:00:00   passwd
```

3. 가상말단(tty1)에서 ctrl+c건을 눌러 passwd응용프로그람을 취소시킵니다.

이 실례에서 /usr/bin/passwd응용프로그람(passwd_exec_t형으로 표식된것임)이 실행될 때 체계관리자의 쉘프로쎄스는 passwd_t령역에로 이행됩니다. 이때 이 형은 프로쎄스들에 대해서는 령역을 정의하며 화일들에 대해서는 형을 정의한다는것을 명심하여야 합니다.

실행되고있는 프로쎄스들에 대한 보안조작체계문맥들을 보려면 ps –eZ지령을 사용하여야 합니다.

```
…
system_u:system_r:local_login_t:s0-s15:c0.c1023   2346  ?  00:00:00   login
root:system_r:vsftpd_t:s0-s15:c0.c1023   3657  ?    00:00:00   vsftpd
system_u:system_r:local_login_t:s0-s15:c0.c1023   4142  ?  00:00:00   login
root:sysadm_r:sysadm_t:s0-s15:c0.c1023   4147  tty2  00:00:00   bash
root:sysadm_r:sysadm_t:s0-s15:c0.c1023   4238  tty1  00:00:00   bash
root:sysadm_r:sysadm_t:s0-s15:c0.c1023   4268  tty1  00:00:00   ps
```

역할 system_r는 체계프로쎄스들 즉 대몬들과 같은것들에 사용됩니다. 그러면 형시행은 매 령역들을 갈라놓게 됩니다.

제 3 절 사용자에 대한 보안문맥

봉사기사용자와 관련된 보안조작체계문맥을 보려면 id –Z지령을 사용하여야 합니다.

다음의 실례는 targeted형보안방책에서의 사용자의 보안문맥을 보여줍니다.

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

이것은 《붉은별》봉사기용체계 3.0 판에서 targeted형보안방책을 시행하는 경우 봉사기사용자들은 기정으로 제한을 받음이 없이 실행된다는것을 보여줍니다..

우의 보안조작체계문맥은 봉사기사용자가 unconfined_u사용자에 대응되여있는것을 보여주며 역할 unconfined_r로 실행되고 또 령역 unconfined_t에서 실행되고있다는것을 보여줍니다. 이 경우 s0-s0 은 MLS범위를 가리키며 s0 이나 같습니다. 사용자가 접근권한을 가지고있는 분류들은 c0.c1023 으로 정의되며 이것은 모든 분류(c0 부터 c1023 까지)를 의미합니다.

다음의 실례는 rss형보안방책에서의 사용자의 보안문맥을 보여줍니다.

```
root:sysadm_r:sysadm_t:s0-s15.c0.c1023
```

이것은 《붉은별》봉사기용체계 3,0 판에서 rss형보안방책을 시행하는 경우 봉사기사용자는 기정으로 체계관리자역할을 할당받아 실행된다는것을 보여줍니다.

우의 보안조작체계문맥은 봉사기사용자가 체계에 기정으로 가입할 때 보안조작체계사용자 root에 대응되여있다는것을 보여주며 역할 sysadm_r로 실행되고 또 령역 sysadm_t에서 실행되고있다는것을 보여줍니다. 이 실례에서 s0-s15 은 MLS범위를 가리키며 사용자가 접근권한을 가지고있는 범주모임은 c0.c1023 으로 정의되여있습니다.

《붉은별》봉사기용체계 3.0 판에서 제공하고있는 targeted형보안방책과 rss형보안방책에 대하여서는 《제 3 장 보안방책의 류형》에서 구체적으로 설명합니다.

제 3 장 보안방책의 류형

제 1 절 targeted형 보안방책

targeted형방책(목표지정형)은 《붉은별》3.0 봉사기용체계에서 사용되는 선택가능한 보안조작체계방책입니다. targeted방책을 사용할 때 접근조종의 대상으로 되는(targeted) 프로쎄스들은 제한을 받는 령역에서 실행되며 targeted가 아닌 프로쎄스들은 제한을 받지 않는 령역에서 실행됩니다. 실례로 기정으로 가입된 사용자들은 령역 unconfined_t에서 실행되며 init에 의하여 기동된 체계프로쎄스들은 령역 initrc_t에서 실행되고 이 령역들은 모두 제한을 받지 않습니다.

제한을 받지 않는 령역들(제한받는 령역들도 마찬가지입니다.)은 실행, 쓰기할수 있는 기억기검사를 진행하는 주동체로 됩니다. 기정으로 제한을 받는 령역에서 실행되고있는 주동체들은 쓰기가능한 기억기를 할당할수 없으며 그것을 실행할수 없습니다. 이렇게 하면 완충기자리넘침공격에 대한 취약점을 감소시킬수 있습니다.

1. 제한을 받는 프로쎄스

망에 대하여 청취하는 거의 모든 봉사들은 《붉은별》봉사기용체계 3.0판에서 제한을 받습니다. 또한 passwd응용프로그람과 같이 봉사기체계의 root사용자로서 실행되며 사용자들을 위한 과제들을 집행하는 대부분의 프로쎄스들은 제한을 받습니다. 어떤 프로쎄스가 제한을 받을 때 그 프로쎄스는 자기자체의 령역에서 실행되게 됩니다. 즉 httpd프로쎄스는 령역 httpd_t에서 실행됩니다. 만일 제한을 받는 프로쎄스가 공격자에 의하여

변이된 경우 보안조작체계방책구성에 따라 공격자의 자원에로의 접근과 공격자에 의해서 일어날수 있는 손상범위가 제한을 받게 됩니다.

다음의 실례는 보안조작체계가 어떻게 Apache HTTP봉사기(httpd)에서 Samba가 사용하는 화일들과 같이 정확한 표식이 붙여지지 않은 화일들을 읽지 못하게 할수 있는가를 보여줍니다. 여기서는 httpd와 wget패키지들이 설치되여있고 보안조작체계의 targeted방책이 사용되고있으며 보안조작체계가 시행방식에서 실행되고있다고 가정합니다.

1) 보안조작체계가 사용가능하며 그것이 시행방식에서 실행되고있는가, targeted방책이 사용되고있는가를 확인하기 위하여 sestatus지령을 실행합니다.

```
$ /usr/sbin/sestatus
SELinux status:                  enabled
SELinuxfs mount:                 /selinux
Current mode:                    enforcing
Mode from config file:           enforcing
Policy version:                  24
Policy from config file:         targeted
```

보안조작체계가 사용가능하면 SELinux status: enabled가 되돌려지며 보안조작체계가 시행방식에서 실행되고있을 때 Current mode: enforcing이 되돌려집니다. 또한 보안조작체계의 targeted방책이 사용되고있으면 Policy from config file: targeted가 되돌려집니다.

2) 봉사기체계의 root사용자로 touch /var/www/html/testfile지령을 실행하여 화일을 하나 만듭니다.

3) ls –Z /var/www/html/testfile지령을 실행하여 보안조작체계문맥을 봅니다.

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/testfile
```

기정으로 봉사기사용자들은 《붉은별》3.0 봉사기용체계에서 제한을 받지 않고 실행되는데 그 리유는 testfile에 보안조작체계의 사용자 unconfined_u로 표식되여있기때문입니다. 프로쎄스들에 대해서는 RBAC가 사용되며 화일들에 대해서는 사용되지 않습니다. 역할들은 화일들에 대하여 의미를 가지지 않으며 화일들(영구보관소와 망화일체계상에 있는 화일들)에 대해서는 일반적인 역할인 object_r역할이 사용됩니다. /proc/등록부아래에 있는 프로쎄스에 관련되는 화일들은 역할 system_r를 사용할수 있습니다. 형 httpd_sys_content_r는 httpd프로쎄스가 이 화일들에 접근할수 있게 합니다.

4) 봉사기체계의 root사용자로서 service httpd start지령을 실행시켜 httpd프로쎄스를 기동합니다.

```
# /sbin/service httpd start
Starting httpd:                                        [ OK ]
```

5) 봉사기사용자가 쓰기권한을 가지고있는 등록부로 가서 지령 wget http://localhost/testfile을 실행합니다. 기정구성을 변경시키지 않는 경우 이 지령은 성공적으로 실행됩니다.

```
--2012-11-06 17:43:01-- http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `testfile'
[ <=>                              ] 0       --.-K/s   in 0s

2012-12 17:43:01 (0.00 B/s) - `testfile' saved [0/0]
```

6) chcon지령은 화일들에 표식을 다시 붙인다. 그러나 이러한 표식변경들은 화일체계의 표식이 다시 붙여질 때에는 없어지게 됩니다. 화일체계에 대한 재표식때에도 남아있는 영구적인 변경을 위해서는 후에 론의하는 semanage지령을 사용하여야 합니다. 봉사기체계의 root사용자로 다음의 지령을 사용하여 그 형을 Samba에 의해서 사용되는 형으로 변경시킵니다.

```
chcon –t samba_share_t /var/www/html/testfile
```

ls –Z /var/www/html/testfile지령을 실행하여 변경사항들을 봅니다.

```
-rw-r--r-- root root unconfined_u:object_r:samba_share_t:s0 /var/www/html
/testfile
```

7) 현재의 자유접근조종허가권한으로 httpd프로쎄스들이 testfile에 접근할 수 있다는것을 명심하여야 합니다. 봉사기사용자가 쓰기접근권한을 가지는 등록부로 가서 지령 wget http://localhost/testfile을 실행합니다. 기정구성을 변경하지 않는한 이 지령은 실패하게 됩니다.

```
--2012-11-06 14:11:23-- http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2012-11-06 14:11:23 ERROR 403: Forbidden.
```

8) 봉사기체계의 root사용자로 지령 rm – i /var/www/html/testfile을 실행하여 testfile을 삭제합니다.

9) httpd를 실행할 필요가 없으면 봉사기체계의 root사용자로 지령 service httpd stop를 실행하여 httpd를 중지시킵니다.

```
# /sbin/service httpd stop
Stopping httpd:                                        [ OK
]
```

이 실례는 보안조작체계에 추가된 추가적인 보안들에 대해서 보여줍니다. 비록 자유접근조종규칙들이 단계 7 에서 httpd프로쎄스가 testfile에 접근할수 있도록 허가하였다 하더라도 그 화일이 httpd프로쎄스가 접근권한을 가지지 못하는 어떤 형으로 표식되여있기때문에 보안조작체계는 접근을 거부하였습니다.

아래에 보여주는것과 비슷한 오유가 /var/log/audit/audit.log에 기록됩니다.

```
type=AVC msg=audit(1220706212.937:70): avc: denied { getattr } for
 pid=1904 comm="httpd" path="/var/www/html/testfile" dev=sda5 ino=247
```

```
576 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:objec
t_r:samba_share_t:s0 tclass=file
type=SYSCALL msg=audit(1220706212.937:70): arch=40000003 syscall=19
6 success=no exit=-13 a0=b9e21da0 a1=bf9581dc a2=555ff4 a3=2008171
items=0 ppid=1902 pid=1904 auid=500 uid=48 gid=48 euid=48 suid=48 f
suid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=1 comm="httpd" exe="/
usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

또한 아래에 보여주는것과 비슷한 오유가 /var/log/httpd/error_log에 기록되게 됩니다.

```
[Wed Dec 06 23:00:54 2012] [error] [client 127.0.0.1] (13)Permission de
nied: access to /testfile denied
```

2. 제한을 받지 않는 프로쎄스

제한받지 않는 프로쎄스들은 제한받지 않는 령역들에서 실행되는데 실례로 init프로그람들은 제한받지 않는 령역 initrc_t에서 실행되고 제한받지 않는 핵심부프로쎄스들은 령역 kernel_t에서 실행되며 제한받지 않는 봉사기사용자들은 령역 unconfined_t에서 실행됩니다. 제한받지 않는 프로쎄스들에 대하여 보안조작체계방책규칙들이 적용되지만 제한되지 않는 령역들에서 실행되는 프로쎄스들이 거의 모든 접근권한을 가지게 하는 방책규칙들이 존재합니다. 제한되지 않는 령역들에서 실행되는 프로쎄스들은 독점적으로 자유접근조종규칙들을 사용하는것으로 되돌아갑니다. 만일 어떤 제한받지 않는 프로쎄스가 변이되면 보안조작체계는 공격자가 체계자원들과 자료들에 대한 접근권한을 획득하는것을 막을수 없습니다. 그러나 물론 자유접근조종규칙들은 여전히 리용됩니다. 보안조작체계는 자유접근조종규칙들우에서의 보안강화해결책이며 자유접근조종규칙들과 교체되지는 않습니다.

다음의 실례는 제한되지 않는 프로쎄스가 실행될 때 어떻게 Apache HTTP봉사기(httpd)가 Samba에 의하여 사용되도록 되여있는 자료들에 접근

할수 있는가를 보여줍니다. 《붉은별》봉사기용체계 3.0판에서 httpd프로세스는 기정으로 령역 httpd_t에서 실행된다는것을 명심하여야 합니다. 이것은 실례에 불과하며 제품에서는 사용되지 않을수 있습니다. 여기서는 httpd, wget, dbus, audit패키지들이 설치되고 보안조작체계의 targeted방책이 사용되며 보안조작체계가 시행방식에서 실행되고있다고 가정합니다.

1) 지령 sestatus를 실행시켜 보안조작체계가 사용가능한가, 그것이 시행방식에서 실행되고있는가, 보안조작체계의 targeted방책이 사용되고있는가를 확인합니다.

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

보안조작체계가 사용가능하면 SELinux status: enabled가 되돌려지며 보안조작체계가 시행방식에서 실행되고있을 때 Current mode: enforcing이 되돌려집니다. 또한 보안조작체계의 targeted방책이 사용되고있으면 Policy from config file: targeted가 되돌려집니다.

2) 봉사기체계의 root사용자로 지령 touch /var/www/html/test2file을 실행하여 화일을 하나 만듭니다.

3) ls –Z /var/www/html/test2file지령을 실행하여 보안조작체계문맥을 봅니다.

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test2file
```

기정으로 봉사기사용자들은 《붉은별》3.0봉사기용체계에서 제한을 받지 않고 실행되는데 그 리유는 test2file에 보안조작체계의 사용자

unconfined_u로 표식되여있기때문입니다. 프로쎄스들에 대해서는 RBAC가 사용되며 화일들에 대해서는 사용되지 않습니다. 역할들은 화일들에 대하여 의미를 가지지 않으며 화일들(영구보관소와 망화일체계상에 있는 화일들)에 대해서는 일반적인 역할인 object_r역할이 사용됩니다. /proc/등록부아래에 있는 프로쎄스에 관련되는 화일들은 역할 system_r를 사용할수 있습니다. 형 httpd_sys_content_r는 httpd프로쎄스가 이 화일들에 접근할수 있게 합니다.

4) chcon지령은 화일들에 표식을 다시 붙인다. 그러나 이러한 표식변경들은 화일체계의 표식이 다시 붙여질 때에는 없어지게 됩니다. 화일체계에 대한 재표식때에도 남아있는 영구적인 변경을 위해서는 후에 론의하는 semanage지령을 사용하여야 합니다. 봉사기체계의 root사용자로 다음의 지령을 사용하여 그 형을 Samba에 의해서 사용되는 형으로 변경시킵니다.

```
chcon –t samba_share_t /var/www/html/test2file
```

ls –Z /var/www/html/test2file지령을 실행하여 변경사항들을 봅니다.

```
-rw-r--r-- root root unconfined_u:object_r:samba_share_t:s0 /var/www/html
/test2file
```

5) 지령 service httpd status를 실행하여 httpd프로쎄스가 실행되고있는가를 확인합니다.

```
$ /sbin/service httpd status
httpd is stopped
```

만일 출력이 우의것과 차이나는 경우 봉사기체계의 root사용자로 지령 service httpd stop를 실행시켜 httpd프로쎄스를 중지시킵니다.

```
# /sbin/service httpd stop
Stopping httpd:                                    [ OK
]
```

6) httpd프로쎄스가 제한을 받지 않고 실행되도록 하기 위하여 봉사기체계의 root사용자로 다음의 지령을 실행시켜 형 /usr/sbin/httpd을 제한받는 령역에로 이행하지 않는 형으로 변경시킵니다.

```
chcon –t unconfined_exec_t /usr/sbin/httpd
```

7) 지령 ls –Z /usr/sbin/httpd를 실행시켜 /usr/sbin/httpd가 형 unconfined_exec_t로 표식되여있는가를 확인합니다.

```
-rwxr-xr-x  root root system_u:object_r:unconfined_exec_t  /usr/sbin/httpd
```

8) 봉사기체계의 root사용자로 지령 service httpd start를 실행시켜 httpd프로쎄스를 기동합니다. httpd가 성공적으로 기동하면 출력은 다음과 같이 될것입니다.

```
# /sbin/service  httpd  start
Starting httpd:                                                    [  OK
]
```

9) 지령 ps –eZ | grep httpd를 실행시켜 httpd가 령역 unconfined_t에서 실행되고있는것을 확인합니다.

```
$ ps -eZ | grep httpd
unconfined_u:system_r:unconfined_t  7721 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7723 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7724 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7725 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7726 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7727 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7728 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7729 ?        00:00:00  httpd
unconfined_u:system_r:unconfined_t  7730 ?        00:00:00  httpd
```

10) 봉사기사용자가 쓰기권한을 가지고있는 등록부로 가서 지령 wget http://localhost/test2file을 실행합니다. 기정구성을 변경시키지 않는 경우 이 지령은 성공적으로 실행됩니다.

```
--2012-05-07 01:41:10--   http://localhost/test2file
```

```
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `test2file.1'
[ <=>                                  ]--.-K/s    in 0s

2012-05-07 01:41:10 (0.00 B/s) - `test2file.1' saved [0/0]
```

비록 httpd프로세스가 형 samba_share_t로 표식된 화일들에 대한 접근권한을 가지고있지 않다고 해도 httpd는 제한을 받지 않는 령역인 unconfined_t에서 실행되고있으며 자유접근조종규칙들을 사용하는데로 되돌아갑니다. 따라서 그것만으로 wget지령이 성공적으로 실행됩니다. httpd가 제한받는 령역인 httpd_t에서 실행되고있으면 wget지령은 실패하게 됩니다.

11) 지령 restorecon은 화일들에 대하여 기정의 보안조작체계문맥을 복귀시켜줍니다. 봉사기체계의 root사용자로서 지령 restorecon –v /usr/sbin/httpd를 실행시켜 /usr/sbin/httpd에 대하여 기정의 보안조작체계문맥을 복귀시킵니다.

```
# /sbin/restorecon -v /usr/sbin/httpd
restorecon reset /usr/sbin/httpd context system_u:object_r:unconfined_notrans_exec_t:s0-
>system_u:object_r:httpd_exec_t:s0
```

지령 ls –Z /usr/sbin/httpd를 실행시켜 /usr/sbin/httpd가 형 httpd_exec_t로 표식되여있는것을 확인합니다.

```
$ ls -Z /usr/sbin/httpd
-rwxr-xr-x  root root system_u:object_r:httpd_exec_t    /usr/sbin/httpd
```

12) 봉사기체계의 root사용자로 지령 /sbin/service httpd restart을 실행시켜 httpd를 재기동시킵니다. 재기동한 후에 지령 ps –eZ | grep httpd를 실행시켜 httpd가 제한받는 령역인 httpd_t에서 실행되고있는가를 확인합니다.

```
# /sbin/service  httpd  restart
Stopping httpd:                                                   [  OK
 ]
Starting httpd:                                                   [  OK
]
# ps -eZ | grep httpd
unconfined_u:system_r:httpd_t     8880 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8882 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8883 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8884 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8885 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8886 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8887 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8888 ?         00:00:00  httpd
unconfined_u:system_r:httpd_t     8889 ?         00:00:00  httpd
```

13) 봉사기체계의 root사용자로 지령 rm –i /var/www/html/test2file을 실행시켜 test2file을 삭제합니다.

14) httpd를 실행할 필요가 없으면 봉사기체계의 root사용자로 지령 service httpd stop를 실행하여 httpd를 중지시킵니다.

```
# /sbin/service  httpd  stop
Stopping httpd:                                                   [  OK
]
```

이 편에 있는 실례들에서는 변이된 제한받는 프로쎄스로부터 자료를 어떻게 보호할수 있는가(보안조작체계에 의해 보호됨)를 보여주며 마찬가지로 변이된 제한받지 않는 프로쎄스로부터 공격자가 어떻게 자료에 더 잘 접근할수 있는가(보안조작체계에 의하여 보호되지 않음)를 보여줍니다.

3. 제한받는 사용자와 제한받지 않는 사용자

모든 봉사기사용자는 보안조작체계방책을 통하여 하나의 보안조작체계사용자에 대응되게 됩니다. 이렇게 하면 봉사기사용자들이 보안조작체계사용자들에게 부여된 제한조건들을 계승하게 됩니다. 이러한 봉사기사용

자대응관계는 봉사기체계의 root사용자로서 지령 semanage login –l을 실행하면 볼수 있습니다.

```
# /usr/sbin/semanage login -l
Login Name                    SELinux User              MLS/MCS Range
__default__                   unconfined_u              s0-s0:c0.c1023
root                          unconfined_u              s0-s0:c0.c1023
system_u                      system_u                  s0-s0:c0.c1023
```

《붉은별》3.0 봉사기용체계에서 봉사기사용자들은 기정으로 보안조작체계의 __default__가입(이것은 사용자 unconfined_t에 대응)에 대응되게 됩니다. 다음의 실례는 기정대응관계를 정의합니다.

```
__default__                   unconfined_u              s0-s0:c0.c1023
```

다음의 실례에서는 새로운 봉사기사용자의 추가와 그 봉사기사용자가 보안조작체계의 사용자 unconfined_u에 대응되는것을 보여줍니다. 여기서는 봉사기체계의 root사용자가 제한을 받지 않고 실행되고있으며 그것이 《붉은별》3.0 봉사기용체계에서 기정으로 동작한다고 가정합니다.

1) 봉사기체계의 root사용자로 지령 /usr/sbin/useradd newuser를 실행시켜 newuser라는 새로운 봉사기사용자를 만듭니다.

2) 봉사기체계의 root사용자로서 지령 passwd newuser를 실행시켜 봉사기체계의 사용자 newuser에 통과암호를 할당합니다.

```
# passwd newuser
Changing password for user newuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

3) 현재의 대화접속에서 탈퇴하고 봉사기체계의 newuser라는 사용자로 가입합니다. 가입할 때 pam_selinux는 봉사기사용자를 어떤 보안조작체계 사용자로 대응(이 경우에는 unconfined_u)시키며 이때 생기게 되는 보안조작체계문맥을 설정합니다. 그다음 이 문맥을 가지고 봉사기사용자의 쉘이

실행되게 됩니다. 지령 id –Z를 실행시키면 봉사기사용자의 문맥을 볼수 있습니다.

```
[newuser@localhost ~]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

4) 봉사기체계의 newuser의 대화접속에서 탈퇴하고 자신의 계산자리(account)로 가입합니다. 만일 봉사기체계의 사용자 newuser를 삭제하려면 봉사기체계의 root사용자로 지령 /usr/sbin/userdel –r newuser를 실행시키며 그의 사용자등록부도 삭제하려면 그 뒤에 사용자등록부를 써줍니다.

제한받는 봉사기사용자들과 제한을 받지 않는 봉사기사용자들은 기억기검사들을 실행하고 쓰기할수 있는 주동체이며 이것 또한 MCS에 의하여 제약(만일 MLS방책이 리용되는 경우 MLS에 의하여서도 제약)되게 됩니다. 만일 제한을 받지 않는 사용자들이 령역 unconfined_t로부터 자체의 제한받는 령역에로 이행할수 있도록 보안조작체계방책에 정의되여있는 그러한 응용프로그람을 실행한다면 제한받지 않는 봉사기사용자들은 또 그 제한받는 령역의 제약조건에 따라갑니다. 이러한것의 보안상 우점은 봉사기사용자가 제한을 받지 않고 실행되고있다고 해도 응용프로그람은 제한을 받게 되여있어 응용프로그람의 결함이 악용되는것을 방책을 가지고 제한할수 있다는것입니다. 주의할점은 이것이 체계를 사용자로부터 보호하지 못한다는것입니다. 대신에 사용자와 체계는 응용프로그람의 결함에 의하여 초래될수 있는 피해로부터 보호됩니다.

《붉은별》봉사기용체계 3.0 판에서는 다음의 표에 보여주는 제한받는 보안조작체계사용자들이 유효합니다.

표 1. 보안조작체계사용자들의 가능성

사용자	령역	X창문체계	su와 sudo	사용자등록부와	망

				/tmp/에서 실행	
user_u	user_t	yes	No	선택적	yes
staff_u	staff_t	yes	sudo만	선택적	yes

• 령역 guest_t, xguest_t, user_t에서 봉사기사용자들은 보안조작체계방책에서 허가하는 경우 사용자ID설정응용프로그람(setuid)만을 실행할수 있습니다(passwd와 같음). 이러한 사용자들은 su와 /usr/bin/sudo와 같은 setuid응용프로그람들을 실행할수 없기때문에 결국 이러한 응용프로그람을 사용하지 못하여 봉사기체계의 root사용자로 될수 없습니다.

• 령역 guest_t의 봉사기사용자들은 망접근권한을 가지고있지 못하며 따라서 말단을 통하여서만 가입할수 있습니다(여기에 ssh가 포함되는데 사용자들은 ssh를 통하여 가입할수 있지만 다른 체계에 접속하는데 ssh를 사용할수 없습니다).

• 기정으로 령역 staff_t의 사용자들은 /usr/bin/sudo와 동반되는 응용프로그람을 실행할 권한을 가지고있지 않다. 이러한 권한은 관리자에 의하여 구성되여야 합니다.

기정으로 령역 guest_t의 봉사기사용자들은 자기의 사용자등록부나 /tmp/등록부에서 응용프로그람들을 실행할수 없는데 이때 사용자들이 쓰기권한을 가지는 등록부들에서 응용프로그람(사용자의 권한을 계승하는 응용프로그람)들을 실행못하게 합니다. 이렇게 하면 결함이 있거나 악의적인 응용프로그람들이 사용자소유의 화일들을 변경하지 못하게 됩니다.

기정으로 령역 user_t와 staff_t의 봉사기사용자들은 자기의 사용자등록부와 /tmp/에서 응용프로그람들을 실행할수 있습니다. 사용자들이 자기의 사

용자등록부와 /tmp/에서 응용프로그람을 실행할수 있게 허가하거나 막아놓는데 대한 정보를 보려면 《제 5 장 제 6 절 사용자가 실행하는 응용프로그람들에 대한 론리형들》을 참고하여야 합니다.

제 2 절 rss형보안방책

rss형보안방책은 《붉은별》봉사기용체계 3.0 판에서 리용되는 기정보안방책입니다. 이 방책에서는 targeted형보안방책에서와는 달리 제한을 받지 않는(unconfined) 관리자나 프로세스가 존재하지 않습니다.

체계에 가입한 관리자들은 unconfined_t령역을 할당받지 않고 관리자의 책임(capability 혹은 권한)에 따르는 령역형을 할당받으며 init에 의해 기동하는 체계프로세스들은 특정한 령역형(실례:Apache인 경우 httpd_t령역)으로 동작합니다.

아래에서는 《제 1 절 targeted형보안방책》에서 제한을 받지 않는 사용자나 프로세스가 rss형보안방책에서는 존재하지 않는다는것을 실례를 들어 보여줍니다.

1. 사용자접근제한

rss형보안방책에서는 RBAC모형에 기초하여 사용자들의 접근권한을 제한합니다.

《붉은별》봉사기용체계 3.0 판에서는 다음과 같은 3 개의 역할을 제공함으로써 체계관리자들에 대한 임무분리(Separation of Duties)를 실현합니다.

- 체계관리자역할

체계관리자역할 sysadm_r는 체계의 구성과 체계지령의 실행을 비롯하여 체계관리과제들을 실행할수 있도록 합니다. 체계관리자역할 sysadm_r는 일반적으로 가장 높은 기밀성준위(SystemHigh)에서 실행합니다.

-보안관리자역할

보안관리자역할 secadm_r는 보안조작체계시행방식의 변경, 보안방책관련지령의 실행을 비롯하여 보안조작체계와 련관된 기능들을 수행할수 있습니다.

-일반관리자역할

이 역할은 sysadm_r, secadm_r역할로 이행할수 있는 권한을 가진 모든 사용자들을 포함합니다. 이 역할은 secadm_r, sysadm_r역할을 가진 사용자의 가입을 막기 위하여 제공되였습니다. 이상의 사용자들은 허가된 역할모임안에 secadm_r, sysadm_r을 포함하고있어야 합니다.

체계관리자가 국부적으로 봉사기체계에 가입하는가, 원격으로 봉사기체계에 가입하는가에 따라 체계는 서로 다른 역할을 봉사기관리자에게 할당합니다.

국부적으로 봉사기체계에 가입하는 경우 체계는 sysadm_r역할을 할당하며 원격으로 봉사기체계에 가입하는 경우 체계는 staff_r역할을 할당합니다.

원격으로 가입하는 경우 체계관리자가 체계를 관리하려면 newrole지령으로 체계관리자역할로 이행하여야 합니다.(그림 2)

```
root@localhost:~
login as: root
root@172.29.88.138's password:
Last login: Mon Dec  3 15:17:59 2012
-bash: /root/.bash_profile: 허가거부
-bash-4.1# id -Z
root:staff_r:staff_t:s0-s15:c0.c1023
-bash-4.1# newrole -r sysadm_r
암호:
[root@localhost ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@localhost ~]#
```

그림 2. putty를 리용한 봉사기체계의 원격가입

그림 2 에서 4 번째행을 보면 정확한 암호를 입력하였다고 해도 체계관리자는 /root등록부안의 .bash_profile에 대한 접근이 불가능하다는 통보문을 현시합니다.

이것은 체계관리자가 ssh를 리용하여 원격으로 가입하는 경우 staff_r역할을 할당받기때문입니다. 이와 같은 기능은 비법사용자가 ssh를 리용하여 사용자인증을 거치지 않고 비법적으로 접근하는 경우 그 권한을 제한시키기 위하여 제공되는 기능이라고 볼수 있습니다.

다음으로 체계관리자역할과 보안관리자역할을 통하여 관리자에 대한 임무분리를 어떻게 실현하는가를 보기로 합시다.

《붉은별》봉사기용체계 3.0 판에서는 관리자의 기본기능을 체계관리와 보안관리로 구분하였으며 이로부터 관리자역할을 2 개의 역할, 즉 체계관리자역할과 보안관리자역할로 분류하였습니다.

체계관리자역할은 체계구성화일에 대한 접근권한, 체계봉사대몬에 대한 관리권한, 체계관리프로그람에 대한 실행권한을 비롯하여 체계를 관리할수 있는 권한들을 할당받고있습니다.

보안관리자역할은 보안조작체계방책구성에 대한 변경과 기록화일들에 대한 읽기권한, 보안조작체계관련지령들에 대한 실행권한들을 할당받고있습니다.

이와 같은 임무분리기능을 다음의 실례를 통하여 확인할수 있습니다.

1. 체계관리자역할로 httpd봉사를 재기동하고 보안방책관리지령(semanage)을 실행합니다.(그림 3)

```
《붉은별》3.0 (봉사기용체계)
핵심부 2.6.32-120727.RSS3.i686 (i686)
localhost login: root
Password:
Last login: Mon Dec  3 15:37:43 on tty2
[root@localhost ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@localhost ~]# service httpd restart
httpd봉사를 중지합니다.                                    [  확인  ]
httpd봉사를 시작합니다.                                    [  확인  ]
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# semanage login -l
/usr/sbin/semanage: 보안방책을 읽을수 없습니다.
[root@localhost ~]# _
```

그림 3. 체계관리자역할의 실행권한

2. 보안관리자역할로 httpd봉사를 재기동하고 보안방책관련지령(semanage)을 실행합니다.(그림 4)

```
《붉은별》3.0 (봉사기용체계)
핵심부 2.6.32-120727.RSS3.i686 (i686)
localhost login: root
Password:
Last login: Mon Dec  3 15:50:33 on tty2
[root@localhost ~]# newrole -r secadm_r
암호:
[root@localhost ~]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@localhost ~]# service httpd restart
env: /etc/init.d/httpd 허가거부
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# semanage login -l

가입사용자이름         SELinux사용자         MLS/MCS범위

__default__            user_u                s0
root                   root                  s0-s15:c0.c1023
system_u               system_u              s0-s15:c0.c1023
[root@localhost ~]# _
```

그림 4. 보안관리자역할의 실행권한

우의 실례를 통하여 체계관리자역할로는 semanage지령을 실행할수 없으며 보안관리자역할로는 service지령을 실행할수 없다는것을 알수 있습니다.

이상과 같이 rss형보안방책에서는 RBAC모형을 리용하여 관리자에 대한 임무분리를 실현하였습니다.

2. 프로쎄스접근제한

rss형보안방책에서는 TE모형에 기초하여 프로쎄스들의 접근권한을 최소권한의 원칙(Principle of Least Privilege)에서 제한합니다.

《제 1 절 targeted형보안방책》에서 설명한 targeted형보안방책에서는 제한을 받지 않는 령역으로서 unconfined_t가 존재하며 이와 같은 령역을 허가

받은 프로세스나 사용자는 강제접근조종을 시행하지 않는 일반적인 리눅스에서의 프로세스나 root사용자와 같은 권한을 가집니다.

rss형보안방책에서는 이와 같은 unconfined_t령역이 존재하지 않습니다.

우의 실례(《제 1 절 targeted형보안방책》)에서 지정한 다음과 같은 지령을 rss형보안방책에서 시험해봅니다.

1. sestatus지령으로 보안조작체계의 시행방식을 확인합니다.

```
《붉은별》3.0 (봉사기용체계)
핵심부 2.6.32-120727.RSS3.i686 (i686)
localhost login: root
Password:
Last login: Mon Dec  3 15:50:37 on tty2
[root@localhost ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        rss
[root@localhost ~]#
```

그림 5. 보안조작체계의 시행방식

2. 체계관리자역할로 /usr/sbin/httpd의 보안문맥을 다음과 같이 변경시킵니다.

   #chcon –t unconfined_exec_t /usr/sbin/httpd

   다음과 같은 통보문이 현시됩니다.

   chcon: `/usr/sbin/httpd’의 보안문맥을 `system_u:object_r:unconfined_exec_t:s0’로 변경할수 없습니다.

3. 보안관리자역할로 /usr/sbin/httpd의 보안문맥을 다음과 같이 변경시킵니다.

   우와 같은 통보문이 현시됩니다.

rss형보안방책에서는 unconfined_exec_t형을 제공하지 않으며 따라서 httpd화일을 unconfined_exec_t형으로 변경할수 없습니다. 즉 httpd봉사대몬을 제한을 받지 않는 unconfined_t형으로 기동할수 없으며 오직 httpd_t형으로만 기동할수 있도록 되여있습니다.

제 4 장 보안조작체계구성과 관리

제 1 절 보안조작체계패키지

보안조작체계는 《붉은별》봉사기용체계 3.0 판에서 설치시에 선택하여야 설치됩니다. 만일 설치된 경우 보안방책형은 기정으로 rss형(strict형)이 되며 보안조작체계는 시행방식으로 됩니다. 보안조작체계 패키지는 다음과 같습니다.

policycoreutils-python:이 패키지는 semanage, audit2allow, audit2why, chcat 와 같은 지령들을 제공합니다.

policycoreutils: restorecon, secon, setfiles, semodule, load_policy, setsebool 과 같은 보안조작체계 관리지령들을 제공합니다.

selinux-policy: SELinux Referenc Policy를 제공합니다. 이것은 하나의 완성된 보안방책이며 다른 방책들의 기초로 사용됩니다. 례하면 targeted 방책.

selinux-policy-[보안방책형]: 보안조작체계 방책들을 제공합니다. 실례로 targeted 형보안방책을 설치하려면 selinux-policy-targeted를 설치하여야 하고 rss형보안방책을 설치하려면 selinux-policy-rss를 설치하여야 합니다.

libselinux-utils: avcstat, getenforce, getsebool, matchpathon, selinuxconlist, se linuxdefcon, selinuxenabled, setenforce, togglesebool 도구들을 제공합니다.

제 2 절 감시 및 기록화일

보안조작체계 거부통보문은 /var/log/audit/audit.log에 기정으로 다음과 같이 씌여있습니다.

```
type=AVC msg=audit(1223024155.684:49): avc: denied { getattr } for pid
=2000 comm="httpd"
path="/var/www/html/file1" dev=dm-0 ino=399185 scontext=unconfined_u:s
ystem_r:httpd_t:s0
tcontext=system_u:object_r:samba_share_t:s0 tclass=file

May 7 18:55:56 localhost setroubleshoot: SELinux is preventing httpd (ht
tpd_t) "getattr"
to /var/www/html/file1 (samba_share_t). For complete SELinux messages.
run sealert -l
de7e30d6-5488-466d-a606-92c9f40d316d
```

대몬들의 자동적인 기동

auditd와 rsyslogd대몬은 체계가 기동할 때 자동적으로 기동됩니다. root사용자가 지령으로 기동할수도 있습니다.

```
#chkconfig --levels 2345 auditd on
#chkconfig --levels 2345 rsyslog on
```

service service-name status지령으로 봉사가 실행중인가를 알수 있습니다. 례를 들면

```
#service auditd status
auditd (pid 1318)를 실행하고있습니다.
```

봉사가 실행중이 아니면(service-name이 중지) service service-name start지령으로 기동할수 있습니다. 이때 사용자는 root사용자이여야 합니다.

```
# /sbin/service auditd start
auditd프로그람을 기동합니다.                              [  확인  ]
```

제 3 절 기본구성화일

/etc/selinux/config화일은 보안조작체계 기본구성화일입니다. 이 화일은 보안조작체계의 방식과 보안방책사용을 조종합니다.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
# rss – policy for RedStar Server3.0
SELINUXTYPE=rss
```

SELINUX=enforcing

SELINUX항목은 보안조작체계시행방식을 설정하며 3 가지 방식 즉 enforcing, permissive, disable을 가지고있습니다. 보안조작체계방책이 시행되면 보안방책규칙에 따라 접근이 거부되며 그 내용이 기록됩니다. Permissive방식으로 사용하면 보안방책은 시행되지 않습니다. 보안조작체계가 부정접근을 막지만 보안조작체계가 시행방식으로 기동하면 거부된것들은 기록됩니다. Disabled방식을 사용하면 보안조작체계는 사용불가능(보안조작체계모듈은 리눅스핵심부에 있지 않다)으로 되며 오직 자유접근조종규칙만을 사용하게 됩니다.

SELINUXTYPE=rss

SELINUXTYPE항목은 보안방책사용을 설정합니다. rss방책은 기정방책입니다. MLS, strict, targeted를 사용하고싶으면 이 항목을 변경시켜야 합니다. rss방책을 사용하려면 selinux-policy-rss패키지가 있어야 하며

/etc/selinux/config화일에 SELINUXTYPE=rss를 설정하고 체계를 재기동하여야 합니다.

제 4 절 보안조작체계의 시행방식설정

/usr/sbin/getenforce 또는 /usr/sbin/sestatus지령을 리용하여 보안조작체계의 상태를 알수 있습니다. getenforce지령은 Enforcing, Permissive, Disabled를 돌려줍니다. getenforce지령은 보안조작체계가 허용(보안조작체계 규칙들이 시행됨)될때 Enforcing을 돌려줍니다.

```
#getenforce
Enforcing
```

getenforce지령은 보안조작체계가 허용되였을때 Permissive를 돌려주지만 보안조작체계방책규칙들은 시행되지 않고 자유접근조종규칙만 사용됩니다. getenforce지령은 보안조작체계가 사용불가능일 때 Disabled를 돌려줍니다.

sestatus지령은 보안조작체계의 상태와 보안방책의 사용상태를 돌려줍니다.

```
#sestatus
SELinux status:              enabled
SELinuxfs mount:                 /selinux
Current mode:                enforcing
Mode from config file:           enforcing
Policy version:              24
Policy from config file:         rss
```

SELinux status: enable은 보안조작체계가 허용되였을 때 돌려지는 값입니다. Current mode: enforcing은 보안조작체계가 시행방식으로 실행중일 때 돌려지는 값입니다. Policy from config file: rss는 보안조작체계가 rss형방책을 사용할 때 돌려지는 값입니다.

1. 보안조작체계 사용가능설정

보안조작체계가 사용불가능일 때 /etc/selinux/config에서 SELINUX항목은 SELINUX=disabled로 되여있습니다.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
# rss – policy for RedStar Server3.0.
SELINUXTYPE=rss
```

getenforce지령을 리용하면 Disabled가 나타난다.

```
#getenforce
Disabled
```

보안조작체계의 사용가능

1. rpm –qa|grep selinux, rpm –q policycoreutils지령을 사용하여 보안조작체계 패키지들이 설치되였는가를 확인합니다. 이 지도서에서는 selinux-policy-rss, selinux-policy-targeted, selinux-policy, libselinux, libselinux-python, libselinux-utils, policycoreutils-python, policycoreutils-newrole이 설치되였다고 가정합니다. 이러한 패키지가 설치되지 않았다면 root사용자로 yum install *패키지명* 지령으로 설치하여야 합니다.

2. 보안조작체계를 사용가능하게 하기 전에 화일체계에서 매 화일에 보안조작체계문맥에 따라 표식을 붙여야 합니다. 그러면 체계가 기동할 때 령역을 제한하여 접근을 보호합니다. 이 보호를 활성화하려면 /etc/selinux/config에 SELinux=permissive를 설정합니다.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
# rss – policy for RedStar Server3.0.
SELINUXTYPE=targeted
```

3. root사용자로서 reboot지령으로 체계를 재기동합니다. 그러면 기동하는 동안에 화일체계는 표식이 붙게 되며 보안조작체계체계에서 표식처리가 진행됩니다.

```
*** Warning -- SELinux targeted policy relabel is required.
*** Relabeling could take a very long time, depending on file
*** system size and speed of hard drives.
****
```

4. 허가방식에서 보안조작체계방책은 시행되지 않습니다. 시행방식으로 절환하기전에 root사용자로 grep “SELInux is preventing” /var/log/messages지령을 실행시켜 확인합니다. 만일 보안조작체계가 마지막 기동시에 활동을 거부시키지 못하면 이 지령은 그 어떤값도 돌려주지 않습니다.

5. /var/log/messages에서 통보문을 거부시키려면 /etc/selinux/config에서 SELinux=enforcing로 설정합니다.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
```

```
# rss – policy for RedStar Server3.0
SELINUXTYPE=rss
```

6. 체계를 재기동한후 getenforce지령으로 확인하면 Enforcing을 돌려준다

```
#getenforce
Enforcing
```

7. 보안관리자(secadm_r역할)로 /usr/sbin/semanage login –l지령을 실행하면 다음과 같은 결과를 보게 됩니다.

```
가입사용자이름      보안조작체계사용자      MLS/MCS범위
__default__         user_u                  s0
root                root                    s0-s15:c0.c1023
system_u            system_u                s0-s15:c0.c1023
```

만일 그렇지 않으면 root사용자로 다음의 지령을 리용하여 사용자넘기기를 진행합니다. 사용자이름에 root, user_u, guest_u로 되여있을수 있습니다.

```
1. #semanage user -a -S rss -P user -R "user_r" -r s0 user_u
2. #semanage user -a -S rss -P user -R "staff_r secadm_r sysadm_r syste
m_r" -r s0-s15:c0.c1023 root
3. #semanage login -m -S rss -s "user_u" -r s0 __default__
4. #semanage login -m -S rss -s "root" -r s0-s15:c0.c1023 root
5. #semanage user -a -S rss -P user -R guest_r guest_u
```

2. 보안조작체계 사용불가능설정

보안조작체계 사용불가능설정은 /etc/selinux/config에서 SELinux=disabled

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
```

```
# mls - Multi Level Security protection.
# rss – policy for RedStar Server3.0.
SELINUXTYPE=rss
```

체계를 재기동한후 getenforce 지령을 주면 Disabled가 됩니다.

```
#getenforce
Disabled
```

제 5 절 론리형값

론리형값목록은 매개의 동작상태를 보여주는데 on, off로 되여있습니다.

semanage boolean –l지령은 root사용자로 실행합니다.

```
#semanage boolean -l
SELinux boolean                    Description
ftp_home_dir         -> off Allow ftp to read and write files in the user h
ome directories
xen_use_nfs          -> off Allow xen to manage nfs files
xguest_connect_network -> on Allow xguest to configure Network Manag
er
```

SELinux boolean렬은 론리형이름입니다. Description렬은 론리형의 on, off와 론리형의 기능을 설명합니다.

다음의 실례에서 ftp_home_dir론리형은 off이며 FTP대몬(vsftp)이 사용자등록부에서 읽기, 쓰기를 하지 못하도록 방지합니다.

```
ftp_home_dir -> off Allow ftp to read and write files in the user home
directories
```

Getsebool –a지령은 전체 론리형을 렬거하며 그것들의 on, off를 보여주지만 그 기능에 대하여서는 설명하지 않습니다.

```
# getsebool -a
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

getsebool ***Boolean-name***지령을 실행하면***Boolean-name***의 상태목록을 보여줍니다..

```
#getsebool  allow_console_login
allow_console_login  --> off
```

공백분리를 사용하면 다중론리형목록으로 목록화합니다.

```
#getsebool  allow_console_login  allow_daemons_dump_core
allow_console_login  --> off
allow_daemons_dump_core  --> on
```

2) 론리형 구성

setsebool ***론리형이름*** **x**지령은 론리형의 on, off를 돌려는데 ***론리형이름***은 론리형의 이름입니다. X는 론리형이 on일때는 on을 돌려주고 그렇지 않을 때에는 off를 돌려줍니다.

다음 실례는 httpd_can_network_connect_db론리형에 대한 구성을 보여줍니다.

1. 기정으로 httpd_can_network_connect_db는 off로 되여있으며 Apache HTTP봉사기스크립트를 막고 자료기지봉사기로부터 모듈화합니다.

```
#getsebool  httpd_can_network_connect_db
httpd_can_network_connect_db  --> off
```

2. 림시적으로 Apache HTTP봉사기스크립트는 허가되여있고 자료기지봉사기와의 련결을 모듈화합니다. Setsebool httpd_can_network_connect_db on지령을 보안관리자(secadm_r역할)로 실행합니다.

3. Getsebool httpd_can_network_connect_db지령을 주면 론리형이 on으로 변화된 내용이 얻어집니다.

```
#getsebool  httpd_can_network_connect_db
httpd_can_network_connect_db  --> on
```

4. 이 변경은 체계를 재기동하여야 합니다. 체계를 재기동하지 않으려면 **setsebool –P *론리형이름* on**지령을 실행합니다.

```
#setsebool -P httpd_can_network_connect_db on
```

5. 림시적으로 원래대로 돌아가려면 보안관리자로 들어가 setsebool httpd_can_network_connect_db off지령을 실행합니다. 체계를 재기동하여도 설정값이 변하지 않도록 하기 위하여서는 setsebool –P httpd_can_network_connect_db off를 실행합니다.

3) NFS와 CIFS에 대한 론리형

기정으로 NFS는 의뢰기측에서 NFS화일체계에 의하여 거부되는 표식이 붙어 탑제됩니다. 공통적인 방책들에서는 기정으로 nfs_t형을 사용합니다. 또한 Samba탑제는 방책상 거부되는 표식이 붙으며 공통적인 방책들에서는 기정으로 cifs_t형을 사용합니다.

Setsebool과 semanage지령은 보안관리자역할(secadm_r역할)만이 실행할수 있습니다. Setsebool –P지령은 변경을 요구하며 재기동을 바라지 안는다면 –P선택항목을 사용하지 않습니다.

Apache HTTP 봉사기

NFS화일체계로 접근을 허가합니다.(화일은 nfs_t형이 붙는다)

```
#setsebool –P httpd_use_nfs on
```

Samba화일체계로 접근을 허가합니다.(화일은 cifs_t형이 붙는다)

```
#setsebool –P httpd_use_cifs on
```

Samba

NFS화일체계를 반입하기

```
#setsebool –P samba_share_nfs on
```

FTP(vsftpd)

NFS화일체계로 접근을 허가

```
#setsebool –P allow_ftpd_use_nfs on
```

Samba화일체계로 접근을 허가

```
#setsebool –P allow_ftpd_use_cifs on
```

다른 봉사들

다른 봉사를 위한 론리형과 관련된 NFS목록

```
#semanage boolean –l | grep nfs
```

다른 봉사를 위한 론리형과 관련된 Samba목록

```
#semanage boolean –l | grep cifs
```

제 6 절 화일문맥설정

보안조작체계가 실행중인 상태에서 모든 프로세스와 화일들은 보안관련 정보로써 표식이 붙게됩니다. 이 정보를 보안조작체계문맥이라고 합니다. 화일에 붙은것을 보려면 ls –Z지령을 리용하면 알수 있습니다.

```
$ ls -Z file1
-rw-rw-r--.   kkh kkh   user_u:object_r:user_home_t:s0          file1
```

이 실례에서 보안조작체계문맥은 사용자(user_u), 역할(object_r), 형(user_home_t), 기밀성준위(s0) 입니다. 이 정보는 접근조종에 리용됩니다. 자유접근조종체계에서 접근은 리눅스사용자와 집단식별자에 의하여 조종됩니다. 보안조작체계규칙은 자유접근조종규칙후에 검사됩니다. 보안조작체계규칙은 자유접근조종규칙이 접근거부되면 사용되지 않습니다.

보안문맥화일을 관리하는 지령들에는 chcon, semanage, fcontext, restorecon과 같은 지령들이 있습니다.

1) 림시변경:chcon

chcon지령은 보안조작체계문맥을 변경시킵니다.

chcon –t ***형 화일이름*** 지령을 실행하면 화일의 형을 변경하는데 ***형***에는 httpd_sys_content_t와 같이 주며 ***화일이름***에는 화일 또는 등록부이름을 줍니다.

chcon –R –t ***형 등록부이름*** 지령을 실행하면 등록부의 형과 그의 문맥을 변경하는데 ***형***은 httpd_sys_content_t처럼주고 ***등록부이름***에는 등록부이름을 줍니다.

화일 또는 등록부형 변경

1. 등록된 사용자 kkh로 가입합니다.

2. touch file1 지령을 실행하여 새화일을 창조합니다. ls –Z file1 지령을 리용하여 file1 의 보안조작체계문맥을 봅니다.

```
$ ls -Z file1
-rw-rw-r--.    kkh  kkh   user_u:object_r:user_home_t:s0          file1
```

이 실례에서 file1 을 위한 보안조작체계문맥은 user_u사용자, object_r역할, user_home_t형, s0 준위를 보여줍니다.

3. chcon –t samba_share_t file1 지령을 리용하여 samba_share_t로 변경시킵니다. 이 지령은 관리자(sysadm_r역할 혹은 secadm_r역할)로 실행되여야 합니다. –t 선택항목은 형만 변경하며 결과를 ls –Z file1 로 확인합니다.

```
$ ls -Z file1
ls: file1 에 접근할수 없습니다. 허가거부
```

4. /sbin/restorecon –v file1 지령을 리용하여 file1 를 위한 보안조작체계 문맥을 복귀합니다. 이 지령은 관리자(sysadm_r역할 혹은 secadm_r역할)로 실행되여야 합니다. –v는 무엇이 변경되였는가를 보여줍니다.

```
#restorecon  -v  /home/kkh/file1
restorecon  reset  file1  context  user_u:object_r:samba_share_t:s0->user_u:object_r:user_home_t:s0
```

이 실례에서 이전의 형은 samba_share_t였는데 user_home_t형으로 저장되였습니다. rss형보안방책을 사용할 때 restorecon지령은 /etc/selinux/rss/contexts/files/등록부에서 화일을 읽습니다.

등록부변경과 내용물형(contents types)

다음과 같이 등록부를 만들고 등록부의 화일형을 Apache HTTP봉사기를 사용하는 형으로 변경시킵니다. 이 실례에서 구성화일은 Apache HTTP봉사기를 사용하려면 /var/www/html/을 사용하게 됩니다.

1. 체계관리자로 mkdir /web지령으로 등록부를 창조합니다. 다음 touch /web/file{1,2,3}지령을 실행시켜 3 개의 빈화일 file1, file2, file3 을 만듭니다. /web/ 등록부와 화일들은 이미 root_t형으로 표식이 붙여집니다.

```
# ls -dZ /web
drwxr-xr-x root root root:object_r:root_t:s0 /web
# ls -lZ /web
-rw-r--r-- root root root:object_r:root_t:s0 file1
-rw-r--r-- root root root:object_r:root_t:s0 file2
-rw-r--r-- root root root:object_r:root_t:s0 file3
```

2. root사용자로 chcon –R –t httpd_sys_content_t /web/지령을 실행하여 /web/ 등록부의 형을 httpd_sys_content_t로 변경시킵니다.

```
# chcon -R -t httpd_sys_content_t /web/
# ls -dZ /web/
drwxr-xr-x root root root:object_r:httpd_sys_content_t:s0 /web/
# ls -lZ /web/
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file1
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file2
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file3
```

3. root사용자로 /sbin/restorecon –R –v /web/지령을 실행하여 보안문맥을 복귀합니다.

```
#restorecon -R -v /web/
restorecon reset /web context root:object_r:httpd_sys_content_t:s0->system_
u:object_r:default_t:s0
```

```
restorecon reset /web/file2 context root:object_r:httpd_sys_content_t:s0->system_u:object_r:default_t:s0
restorecon reset /web/file3 context root:object_r:httpd_sys_content_t:s0->system_u:object_r:default_t:s0
restorecon reset /web/file1 context root:object_r:httpd_sys_content_t:s0->system_u:object_r:default_t:s0
```

2) 일관성변경(Persistent Changes):semanage fcontext

/usr/sbin/semanage fcontext지령은 화일에 대한 보안조작체계문맥을 변경시킵니다. Targeted방책을 사용할 때 이 지령은 만일 file_contexts에 존재하는 화일을 변경하거나 file_contexts.local을 새로운 화일과 등록부(례하면 /web/등록부와 같은)에 추가하려면 /etc/selinux/rss/context/files/file_contexts화일에 추가합니다. 화일체계와 련관되는 보안조작체계문맥설정시에는 setfiles지령을 사용하고 보안조작체계문맥을 복귀하는데는 /sbin/restorecon 지령을 사용합니다. 이것은 /usr/sbin/semanage fcontext이 일관성을 보장해준다는것을 의미합니다. 보안조작체계방책은 어떤 주어진 화일에 대하여 사용자도 변경을 할수 있게 조종합니다.

빠른 참조

보안조작체계문맥을 작성하는데서 화일체계 표식도 변경할수 있습니다.

1. /usr/sbin/semanage fcontext –a ***선택항목 화일이름|등록부이름*** 지령을 실행하는데 화일이나 등록부경로는 전체 경로로 합니다.

2. /sbin/restorecon –v ***화일이름|등록부이름*** 지령을 리용하여 보안문맥을 변경시킵니다.

화일형 변경

화일형을 변경하는것은 다음과 같이 합니다.

1. 체계관리자(sysadm_r역할)로 touch /etc/file1 지령을 실행하여 새 화일을 창조합니다. 기정적으로 새로 만들어지는 화일은 /etc/등록부에 etc_t형으로 표식이 붙습니다.

```
# ls -Z /etc/file1
-rw-r--r-- root root root:object_r:etc_t:s0 /etc/file1
```

2. 보안관리자(secadm_r역할)로 semanage fcontext –a –t samba_share_t /etc/file1 지령을 실행하여 file1 의 형을 samba_share_t형으로 변경하며 –a 선택항목은 새로운 레코드를 추가하며 –t항목은 형(samba_share_t)을 정의합니다. 이 지령들은 직접 형을 변경시키지는 않으며 –file1 는 아직도 etc_t형으로 표식이 붙어있습니다.

```
#semanage fcontext -a -t samba_share_t /etc/file1
# ls -Z /etc/file1
-rw-r--r-- root root root:object_r:etc_t:s0 /etc/file1
```

/usr/sbin/semanage fcontext –a –t samba_share_t /etc/file1 지령은 다음의 입구점을 /etc/selinux/targeted/contexts/files/file_contexts.local로 추가합니다.

```
/etc/file1 unconfined_u:object_r:samba_share_t:s0
```

3. 체계관리자(sysadm_r역할)나 보안관리자(secadm_r역할)로 restorecon –v /etc/file1 지령을 실행하여 형을 변경시킵니다. semanage지령은 /etc/file1 를 file.contexts.local로 추가하며 restorecon지령은 형을 samaba_share_t로 변경시킵니다.

```
#restorecon -v /etc/file1
restorecon reset /etc/file1 context root:object_r:etc_t:s0->system_u:object_r:s
amba_share_t:s0
```

4. 체계관리자(sysadm_r역할)로 rm –i /etc/file1 지령을 실행하여 file1 를 삭제합니다.

5. 보안관리자(secadm_r역할)로 semanage fcontext –d /etc/file1 지령을 실행하여 추가된 /etc/file1 를 삭제합니다. 문맥을 삭제할때 restorecon은 samba_share_t이 아니라 etc_t로 변경시킵니다.

등록부형 변경

새 등록부를 만들고 등록부의 형을 변경하는것을 실례로 고찰하되 형은 Apache HTTP봉사기로 합니다.

1. 체계관리자로 mkdir /web지령을 실행하여 새 등록부를 만듭니다. 이 등록부는 root_t형으로 표식붙습니다.

```
# ls -dZ /web
drwxr-xr-x root root root:object_r:root_t:s0 /web
```

ls –d선택항목은 ls지령에 대하여 등록부만의 정보를 보는것입니다. –Z선택항목은 ls지령을 보안조작체계문맥으로 표시합니다.(실례로 root:object_r:root_t:s0)

2. 보안관리자로 semanage fcontext –a –t httpd_sys_context /web지령을 실행하여 /web/형을 httpd_sys_content_t로 변경시킵니다. –a선택항목은 새 레코드를 추가하며 –t선택항목은 형(httpd_sys_content_t)을 정의합니다. 등록부 /web/의 형은 여전히 default_t로 표식이 붙어있습니다.

```
#semanage fcontext -a -t httpd_sys_content_t /web
#ls -dZ /web
drwxr-xr-x root root root:object_r:root_t:s0 /web
```

semanage fcontext –a –t httpd_sys_content_t /web지령은 다음의 /etc/selinux/rss/contexts/files/file_contexts.local입구점을 추가합니다.

```
/web system_u:object_r:httpd_sys_content_t:s0
```

3. 체계관리자나 보안관리자로 restorecon –v /web지령을 실행하여 형을 변경시킵니다. semanage지령이 /web/를 file.contexts.local로 추가한 후 restorecon 지령은 httpd_sys_content_t형으로 변경합니다.

```
#restorecon -v /web
restorecon reset /web context root:object_r:root_t:s0->system_u:object_r:http
d_sys_content_t:s0
```

기정으로 새롭게 화일을 창조하는 경우 창조되는 화일은 어미등록부의 보안조작체계형을 계승합니다. 이 실례를 리용할 때 보안조작체계문맥을 삭제하기 전에 /web/를 추가시키면 /web/등록부안의 화일과 등록부들은 httpd_sys_content_t로 표식붙여집니다.

4. 보안관리자로 semanage fcontext –d /web지령을 실행하여 /web/문맥을 삭제합니다.

5. 체계관리자나 보안관리자로 restorecon –v /web지령을 실행하여 기정 보안조작체계문맥을 복귀합니다.

등록부와 내용물형을 변경

다음의 실례는 새 등록부를 창조하고 등록부의 형을 Apach HTTP봉사기에서 사용할수 있게 만듭니다. 이 실례는 Apache HTTP봉사기에서 사용하는 등록부(/var/www/html)와 다른데서 작성합니다.

1. 체계관리자로 mkdir /web지령을 실행하여 새로운 등록부를 창조하며 다음 touch /web/file{1,2,3}지령으로 3개의 빈 화일(file1, file2, file3)을 만듭니다. /web/등록부화 화일들은 root_t형으로 표식이 붙어있게 됩니다.

```
# ls -dZ /web
drwxr-xr-x root root root:object_r:root_t:s0 /web
# ls -lZ /web
-rw-r--r-- root root root:object_r:root_t:s0 file1
-rw-r--r-- root root root:object_r:root_t:s0 file2
-rw-r--r-- root root root:object_r:root_t:s0 file3
```

2. 보안관리자로 semanage fcontext –a –t httpd_sys_content_t "/web/(/.*)?"지령을 실행하여 /web/등록부와 그안의 화일들의 형을 httpd_sys_content_t로 변경시킵니다. –a선택항목은 새로운 레코드를 추가하며 –t 선택항목은 형(httpd_sys_content_t)를 정의합니다. "/web/(/.*)?"정규표현식은 semanage지령이 /web/등록부를 변경한다는 의미입니다.

```
# ls -dZ /web
drwxr-xr-x root root root:object_r:root_t:s0 /web
# ls -lZ /web
-rw-r--r-- root root root:object_r:root_t:s0 file1
-rw-r--r-- root root root:object_r:root_t:s0 file2
-rw-r--r-- root root root:object_r:root_t:s0 file3
```

semanage fcontext –a –t httpd_sys_content_t "/web(/.*)?"지령은 다음의 /etc/selinux/rss/contexts/files/file_contexts.local입구점을 추가합니다.

```
/web(/.*)? system_u:object_r:httpd_sys_content_t:s0
```

3. 체계관리자로 restorecon –R –v /web지령을 실행하여 /web/등록부의 형을 변경시키는데 그 등록부안의 화일도 같이 변합니다. –R는 재귀적인 실행을 진행하며 이 지령은 /web/등록부의 화일에 httpd_sys_content_t표식을 붙인다. semanage지령으로 /web(/.*)?에 의하여 file.contexts.local을 추가한 후 restorecon지령 형을 https_sys_content_t로 변경시킵니다.

```
#restorecon -R -v /web
restorecon reset /web context root:object_r:root_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file2 context root:object_r:root_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file3 context root:object_r:root_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file1 context root:object_r:root_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

4. 보안관리자로 semanage fcontext –d "/web(/.*)?"지령을 실행하여 "/web(/.*)?"에 의하여 추가된 문맥들을 삭제합니다.

5. 체계관리자로 restorecon –R –v /web지령을 실행하여 보안조작체계 문맥을 복귀합니다.

문맥의 삭제와 추가

1. 보안관리자로 semanage fcontext –a –t httpd_sys_content_t /test지령을 실행합니다. /test/등록부는 존재하지 않습니다. 이 지령은 /etc/selinux/rss/contexts/files/file_contexts.local문맥을 추가합니다.

```
/test   system_u:object_r:httpd_sys_content_t:s0
```

2. 문맥을 삭제하기 위하여 보안관리자로 semanage fcontext –d *화일이름|등록부이름* 지령을 실행하는데 여기서 *화일이름|등록부이름* 은 file_contexts.local에서의 첫번째 부분입니다. 다음의 실례는 file_contexts.local에서의 문맥실례입니다.

```
/test   system_u:object_r:httpd_sys_content_t:s0
```

제 7 절 file_t와 default_t형

확장된 속성을 지원하는 화일체계에서 디스크에서 보안조작체계 문맥이 부족한 화일이 접근할때 보안조작체계 방책으로 기정적인 문맥을 가지고 있으면 승인해줍니다. 공동 방책들에서는 기정 문맥들은 file_t형을 사용합니다. 이것만 사용하게 한다면 디스크상에서 문맥이 없는 화일은 방책에서 구별할수 있으며 일반적으로 제한받는 령역으로 접근할수 없게됩니다. file_t형은 정확하게 표식이 붙는 화일체계에 존재하지 않을수 있는데 그것은 체계에서 모든 화일들은 보안조작체계가 실행될때 보안문맥을 가지고있기 때문이며 file_t형은 화일문맥구성화일(file-context configuration:/etc/selinux/targeted/context/files/에서 있는 화일들은 화일과 등록부들에 대한 문맥을 정의하고있습니다. 이 등록부의 화일들은 restorecon

로 읽을수도 있고 setfilesto에 의하여 기정문맥을 회복할수도 있습니다)에서 결코 사용되지 않습니다. default_t형은 화일문맥구성화일((file-context configuration)에서 어떤 다른 패턴과 일치되지 않는 화일에서 사용되므로 화일과 같은것들은 디스크에서 문맥을 가지고있지 않는 화일로부터 식별될수 있고 일반적으로 제한받는 령역으로의 접근이 거절됩니다. 높은준위의 등록부를 /mydirectory/로 새로 만든다면 이등록부는 default_t형으로 표식이 붙을수 있습니다. 봉사가 등록부에 접근할 필요가 있다면 이 배치는 화일문맥구성화일을 갱신합니다.

제 8 절 보안조작체계 표식보존

이 절에서는 체계안의 화일과 등록부를 복사, 이동시킬 때 보안조작체계표식과 관련하여 어떤 문제가 발생하는가를 서술합니다. 또한 복사할 때 문맥을 어떻게 보존하는가를 서술합니다.

1. 화일과 등록부복사

화일 또는 등록부를 복사할 때 새로운 화일 또는 등록부가 만들어집니다. 새 화일이나 등록부의 문맥은 기정표식할당규칙에 따라 결정되며 초기의 화일이나 등록부의 문맥으로 하지 않습니다. 실례로 사용자등록부에서 만들어진 화일들은 admin_home_t형으로 표식이 붙습니다.

```
#touch  file1
#ls  -Z  file1
-rw-r--r-- root  root  root:object_r:admin_home_t:s0   file1
```

만일 다른 등록부 즉 례하면 /etc/로 복사한다면 새 화일은 /ect/등록부에 기정표식할당규칙이 적용되지 않고 만들어집니다. 복사된 화일은 초기의 문맥을 가지고있지 않습니다.

```
#ls -Z file1
-rw-r--r-- root root root:object_r:admin_home_t:s0 file1
# cp file1 /etc/
$ ls -Z /etc/file1
-rw-r--r-- root root root:object_r:etc_t:s0 /etc/file1
```

file1가 /etc/로 복사될 때 만일 /etc/file1가 존재하지 않는다면 /etc/file1는 새 화일로 만들어집니다. 실례에서 보는 바와 같이 /etc/file1는 etc_t형으로 표식붙어집니다.

화일이 이미 존재하는 경우 복사할 때 존재하는 화일의 문맥은 보존되며 만일 사용자가 cp지령에서 선택항목을 사용하지 않으면 초기화일의 보안문맥이 –preserve=context로 보존됩니다. 보안조작체계규칙은 복사하는 과정에 문맥을 보존할수 있습니다.

보안조작체계문맥없이 복사

cp지령으로 화일을 복사할 때 선택항목이 없으면 형은 targeted, 어미등록부로부터 계승됩니다.

```
#touch file1
#ls -Z file1
-rw-r--r-- root root root:object_r:admin_home_t:s0 file1
#ls -dZ /var/www/html/
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
# cp file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 /var/www/html/file1
```

이 실례에서 file1는 사용자등록부에 만들어지며 admin_home_t형으로 표식을 붙인다. /var/www/html/등록부는 httpd_sys_content_t형으로 표식이 붙습니다. ls –dZ /var/www/html/지령으로 알수 있습니다. file1가 /var/www/html/로

복사할 때 httpd_sys_content_t형으로 계승되는데 ls –Z /var/www/html/file1지령으로 알아볼수 있습니다.

복사할 때 보안조작체계 문맥보존

cp –preserve=context지령을 사용하여 복사할 때 문맥을 보존합니다.

```
#touch  file1
#ls  -Z  file1
-rw-r--r-- root  root  root:object_r:admin_home_t:s0   file1
#ls  -dZ  /var/www/html/
drwxr-xr-x  root  root  system_u:object_r:httpd_sys_content_t:s0  /var/www/html/
# cp  --preserve=context   file1   /var/www/html/
# ls  -Z  /var/www/html/file1
-rw-r--r-- root  root  root:object_r:admin_home_t:s0   /var/www/html/file1
```

이 실례에서 file1는 사용자등록부에 만들어지며 admin_home_t형으로 표식이 붙습니다. /var/www/html/등록부는 httpd_sys_content_t형으로 표식이 붙으며 ls –dZ /var/www/html/지령을 리용하여 알수 있습니다. –preserve=context선택항목을 사용하여 복사하는 과정에 보안조작체계문맥을 보존하며 그 결과는 ls –Z /var/www/html/file1지령으로 알수 있고 file1의 admin_home_t형은 /var/www/html/로 복사할 때 보존됩니다.

문맥의 복사와 변경

cp –Z지령을 사용하여 목적하는 문맥을 변경합니다. 다음 실례는 사용자등록부에서 실행된것입니다.

```
#touch  file1
#cp  -Z  system_u:object_r:samba_share_t:s0   file1   file2
#ls  -Z  file1   file2
-rw-r--r-- root  root  root:object_r:admin_home_t:s0   file1
-rw-r--r-- root  root  system_u:object_r:samba_share_t:s0   file2
#rm file1  file2
```

이 실례에서 문맥은 –Z선택항목으로 정의합니다. –Z항목이 없으면 file2은 root:object_r:admin_home_t문맥으로 표식이 붙을것입니다.

화일덧쓰기

화일을 덧쓰기할 때 보안문맥은 보존됩니다. 실례를 들면

```
# touch /etc/file1
# ls -Z /etc/file1
-rw-r--r-- root root root:object_r:etc_t:s0 /etc/file1
# touch /tmp/file2
# ls -Z /tmp/file2
-rw-r--r-- root root root_u:object_r:user_tmp_t:s0 /tmp/file2
# cp /tmp/file2 /etc/file1
# ls -Z /etc/file1
-rw-r--r-- root root root:object_r:etc_t:s0 /etc/file1
```

이 실례에서 두 화일이 복사되였습니다. /etc/file1는 etc_t형으로 표식붙고 /tmp/file2은 user_tmp_t형으로 표식이 붙습니다. cp /tmp/file2 /etc/file1지령은 file1를 file2로 복사합니다. 복사후에 ls –Z /etc/file1지령은 file1는 etc_t형으로 표식이 붙는것을 알수 있습니다.

2. 화일과 등록부의 이동

화일과 등록부들은 이동할 때 현재의 보안조작체계문맥을 보존합니다. 많은 경우 이동할 때 정확히 할당되지는 않습니다. 다음의 실례에서 사용자등록부로부터 /var/www/html/로 이동하는것을 보여주는데 Apache HTTP 봉사기로 사용할수 있게 합니다. 화일을 이동한 후 정확한 보안문맥은 계승되지 않습니다.

1. cd지령을 파라메터가 없이 실행하여 사용자등록부로 변경합니다. 사용자등록부에서 touch file1지령을 실행하여 화일을 만듭니다. 이 화일은 user_home_t형 표식이 붙습니다.

```
$ ls -Z file1
-rw-rw-r-- user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

2. ls –dZ /var/www/html지령을 실행하여 /var/www/html/등록부의 보안문맥

을 볼수 있습니다.

```
#ls -dZ /var/www/html/
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t:s0 /var/www/ht
ml/
```

기정으로 /var/www/html/등록부는 httpd_sys_content_t형으로 표식붙습니다. 화일과 등록부는 /var/www/html/등록부에 이 형을 계승하여 만들어지며 마찬가지로 이러한 형으로 표식이 붙습니다.

3. 체계관리자로 mv file1 /var/www/html/지령을 실행하여 file1를 /var/www/html/등록부로 이동합니다. 이 화일이 이동한 후 file1의 형은 user_home_t형으로 보존됩니다.

```
# mv file1 /var/www/html/
# ls -Z /var/www/html/file1
-rw-rw-r-- user1 group1 unconfined_u:object_r:user_home_t:s0 /var/www/ht
ml/file1
```

기정으로 Apache HTTP봉사기는 user_home_t형이 붙은 화일을 읽을수 없습니다. 만일 웨브페지를 포함한 모든 화일이 user_home_t형으로 표식이 붙거나 Apache HTTP봉사기가 읽을수 없는 다른 형으로 되여있으면 웨브열람기나 본문기초의 Web열람기로의 접근을 금지시킵니다.

3. 기정보안조작체계문맥을 검사

화일들과 등록부들이 정확한 보안조작체계문맥을 가지고있는가 matchpathcon지령으로 검사할수 있습니다(matchpathcon(8)안내페지를 참고). 다음의 실례는 matchpathcon지령을 리용하여 /var/www/html/등록부에서 정확히 표식이 붙었는가를 보여줍니다.

1. 체계관리자로 touch /var/www/html/file{1,2,3}지령을 실행하여 3개의 화일(file1, file2, file3)을 만듭니다. 이러한 화일들은 /var/www/html/등

록부로부터 httpd_sys_content_t형을 계승합니다.

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file1
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file2
-rw-r--r-- root root root:object_r:httpd_sys_content_t:s0 file3
```

2. 체계관리자로 chcon –t samba_share_t /var/www/html/file1지령을 실행하여 file1의 형을 samba_share_t로 변경시킵니다. 주의할것은 Apache HTTP봉사기는 samba_share_t형으로 표식이 붙은 화일이나 등록부를 읽을수 없습니다.

3. matchpathcon –V선행항목으로 보안조작체계의 기존방책과 현재 보안조작체계의 방책을 비교합니다. matchpathcon –V /var/www/html/*지령을 실행하여 /var/www/html/등록부안에 있는 모든 화일을 검사합니다.

```
#matchpathcon -V /var/www/html/*
/var/www/html/file1 has context root:object_r:samba_share_t:s0, should be system_u:object_r:httpd_sys_content_t:s0
/var/www/html/file2 verified.
/var/www/html/file3 verified.
```

다음의 결과는 matchpathcon지령으로 file1가 samba_share_t형으로 표식이 붙었다는것을 보여주지만 httpd_sys_content_t형으로 표식이 붙어야 한다는 것을 보여줍니다.

```
/var/www/html/file1 has context root:object_r:samba_share_t:s0, should be system_u:object_r:httpd_sys_content_t:s0
```

표식문제를 해결하기 위하여 또 Apache HTTP봉사기가 file1에 접근하기 위하여 체계관리자로 matchpathcon –v /var/www/html/file1지령을 실행합니다.

```
#restorecon -v /var/www/html/file1
restorecon reset /var/www/html/file1 context root:object_r:samba_share_t:s0
-
>system_u:object_r:httpd_sys_content_t:s0
```

4. tar를 리용한 화일보존

tar는 기정으로 확장속성들을 보존유지하지 않습니다. 확장속성들에 보안조작체계문맥들이 보관되기때문에 이 문맥들이 화일보존시에 잃어질수도 있습니다. tar –selinux를 사용하여 이 문맥들을 보존유지하는 보존물들을 만들수 있습니다. 만일 어떤 tar보존물이 확장속성이 없는 화일들을 포함하고있거나 혹은 사용자가 확장속성을 체계기정설정과 같이 하려면 restorecon을 거쳐 보존물을 풀어놓습니다.

```
#tar -xvf archive.tar | /sbin/restorecon -f –
```

다음의 실례에서는 보안조작체계문맥들을 보존유지하는 tar보존물을 만드는 과정을 보여줍니다.

1. 체계관리자로 지령 touch /var/www/html/file{1,2,3}를 실행하여 세개의 화일(file1, file2, file3) 을 만듭니다. 이 화일들은 /var/www/html/등록부로부터 형 httpd_sys_content_t을 계승합니다.

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--   root root root:object_r:httpd_sys_content_t:s0   file1
-rw-r--r--   root root root:object_r:httpd_sys_content_t:s0   file2
-rw-r--r--   root root root:object_r:httpd_sys_content_t:s0   file3
```

2. 지령 cd /var/www/html/을 실행하여 /var/www/html/등록부로 갑니다. 이 등록부에서 체계관리자로 지령 tar --selinux -cf test.tar file{1,2,3}을 실행하여 test.tar라고 불리우는 Tar보존물을 만듭니다.

3. 체계관리자로서 지령 mkdir /test를 실행하여 새 등록부를 하나 만들고 지령 chmod 777 /test/를 실행하여 이 /test/등록부에 모든 사용자들이 충분히 접근할수 있게 합니다.

4. 지령 cp /var/www/html/test.tar /test/를 실행하여 /test/등록부안에 test.tar 화일을 복사합니다.

5. 지령 cd /test/를 실행하여 /test/등록부에 갑니다. 이 등록부에서 지령 tar -xvf test.tar를 실행하여 Tar보존물을 푼다.

6. 지령 ls –lZ /test/를 실행하여 보안조작체계문맥을 확인합니다. --selinux가 사용되지 않았을 때 나타나는 default_t로 되지 않고 형 httpd_sys_content_t이 보존유지된것을 알수 있습니다.

```
#ls -lZ /test/
-rw-r--r--   root  root  root:object_r:httpd_sys_content_t:s0   file1
-rw-r--r--   root  root  root:object_r:httpd_sys_content_t:s0   file2
-rw-r--r--   root  root  root:object_r:httpd_sys_content_t:s0   file3
-rw-r--r--   root  root  root:object_r:root_t:s0   test.tar
```

7. /test/등록부가 더는 필요가 없는 경우 체계관리자로서 지령 rm –ri /test/을 실행하여 등록부와 그 안의 화일들을 삭제할수 있습니다.

모든 확장속성들을 보존유지하는 추가선택 --xattrs과 같이 tar에 대한 자세한 내용을 보려면 tar(1)의 안내페지를 참고하여야 합니다.

제 9 절 정보수집도구

이 도구들은 지령행도구들로서 형식화된 출력을 줍니다. 이것들을 지령행관흐름처리의 일부분으로 사용하기는 좀 까다롭기는 하지만 고속으로 정보를 수집하고 형식화된 정보를 제공하는 우점이 있습니다.

avcstat

이 지령은 기동시각부터 접근벡토르의 완충물에 대한 통계량을 간단히 출력해줍니다. 사용자는 초단위로 시간간격을 설정해주어 실시간적으로 통계량들을 감시할수 있습니다. 이 지령은 처음에 출력한 이후로 통계량들을 갱신시켜줍니다. 사용되는 통계량화일은 /selinux/avc/cache_stats이며 선택항목 "–f 화일의 경로"를 리용하여 다른 완충기억화일을 지적할수 있습니다.

```
[root@localhost ~]# avcstat
lookups      hits        misses    allocs    reclaims    frees
47517410     47504630    12780     12780     12176       12275
```

seinfo(이 지령은 현재《붉은별》봉사기용체계3.0판에는 설치되여있지 않으므로 참고적으로 볼것,setools패키지에 있음)

이 지령은 클라스, 형, 론리형, 허가규칙 등과 같은 방책의 세부적인 정보들을 서술하는데 쓰인다. seinfo는 입구값으로서는 policy.conf화일이나 2진방책화일을 사용하는 지령행편의프로그람입니다.

seinfo의 출력은 2진화일과 원천화일사이에서 가변적입니다. 실례로 방책원천화일에서는 여러개의 규칙요소들을 간단히 한행으로 묶어놓기 위해서 인용부호 {}를 사용합니다. 이러한 방법을 속성들에서도 리용할수 있는데 거기서는 하나의 간단한 속성을 하나이상의 형으로 확장합니다. 이것들이 확장되여 2 진방책화일에서 더는 관련이 없는것으로 되기때문에 탐색결과들에서 되돌림값 0 을 가집니다. 그러나 이전에 인용괄호를 사용하였던 매 한행의 규칙들이 수많은 개별적인 행들로 되기때문에 규칙의 수가 급격히 증가하게 됩니다.

일부 항목들은 2 진방책에 제공되지 않습니다. 실례로 neverallow규칙들은 방책을 콤파일하는동안에만 검사되고 실행시간에는 검사되지 않으며 초기의 SID들은 체계기동시 핵심부에서 적재하는 방책보다 먼저 요구되기때문에 2진화일의 부분이 아니다.

```
[root@localhost ]# seinfo
Statistics for policy file: /etc/selinux/targeted/policy/policy.24
Policy Version & Type: v.24 (binary, mls)
   Classes:            77    Permissions:       229
   Sensitivities:       1    Categories:       1024
   Types:            3001    Attributes:        244
   Users:               9    Roles:              13
   Booleans:          158    Cond. Expr.:       193
   Allow:          262796    Neverallow:          0
   Auditallow:         44    Dontaudit:      156710
```

```
    Type_trans:          10760       Type_change:            38
    Type_member:             44       Role  allow:             20
    Role_trans:             237       Range_trans:          2546
    Constraints:             62      Validatetrans:           0
    Initial   SIDs:          27      Fs_use:                 22
    Genfscon:                82       Portcon:              373
    Netifcon:                 0       Nodecon:                0
    Permissives:             22      Polcap:                  2
[root@localhost  ]#
```

seinfo지령은 또한 령역속성을 가진 형의 총개수를 목록화하며 여러개의 제한받는 프로쎄스들의 총개수의 추정값을 줍니다.

```
# seinfo  -adomain  -x |  wc -l
550
```

모든 령역형들이 다 제한받는것은 아니다. 제한받지 않는 령역들을 보려면 unconfined_domain속성을 사용하여야 합니다.

```
# seinfo  -aunconfined_domain_type  -x | wc -l
52
```

허가방식의 령역들은 추가선택 --permissive을 리용하여 계수됩니다.

```
# seinfo  --permissive  -x | wc -l
31
```

우의 지령에서 | wc –l부분을 삭제하면 모든 목록을 볼수 있습니다.

sesearch(이 지령은 현재 《붉은별》봉사기용체계 3.0 판에는 설치되여있지 않으므로 참고적으로 볼것,setools패키지에 있음)

사용자는 sesearch지령을 사용하여 방책에서 특별한 형에 대하여 탐색할수 있습니다. 이를 위해서 방책원천화일이나 2 진화일에서 검색할수 있습니다. 아래에 실례를 보여줍니다.

```
[scott@localhost ~]$ sesearch --role_allow -t httpd_sys_content_t \ /etc/seli
nux/targeted/
policy/policy.24
Found  20  role  allow  rules:
```

```
    allow  system_r  sysadm_r;
    allow  sysadm_r  system_r;
    allow  sysadm_r  staff_r;
    allow  sysadm_r  user_r;
    allow  system_r  git_shell_r;
    allow  system_r  guest_r;
    allow  logadm_r  system_r;
    allow  system_r  logadm_r;
    allow  system_r  nx_server_r;
    allow  system_r  staff_r;
    allow  staff_r  logadm_r;
    allow  staff_r  sysadm_r;
    allow  staff_r  unconfined_r;
    allow  staff_r  webadm_r;
    allow  unconfined_r  system_r;
    allow  system_r  unconfined_r;
    allow  system_r  user_r;
    allow  webadm_r  system_r;
  allow  system_r  webadm_r;
  allow  system_r  xguest_r;
```

sesearch지령은 허가(allow)규칙의 개수를 보여줍니다.

```
# sesearch --allow | wc -l
262798
```

또한 dontaudit규칙의 개수도 보여줍니다.

```
# sesearch --dontaudit | wc -l
156712
```

제 5장 사용자에 대한 제한 부여

《붉은별》봉사기용체계 3.0판은 많은 제한받는 사용자들을 리용할수 있습니다. 매 봉사기사용자는 보안조작체계방책을 통하여 어떤 보안조작체계사용자에 대응되게 되는데 이때 봉사기사용자들은 보안조작체계사용자들에게 부여된 제약조건들을 계승하게 되며 실례로(사용자에 따라) 망을 리용할수 없거나 setuid응용프로그람들을 실행할수 없(보안조작체계방책이 허가하지 않는 한)으며 혹은 su와 sudo지령들을 실행시킬수 없게 됩니다. 이렇게 하면 사용자로부터 체계를 보호할수 있습니다.

제 1절 봉사기사용자와 보안조작체계사용자의 대응관계

보안관리자로서 지령 semanage login –l을 실행시켜 봉사기사용자들과 보안조작체계사용자들사이의 대응관계를 볼수 있습니다.

#semanage login -l

가입사용자이름	보안조작체계사용자	MLS/MCS범위
__default__	user_u	s0
root	root	s0-s15:c0.c1023
system_u	system_u	s0-s15:c0.c1023

《붉은별》봉사기용체계 3.0판에서 봉사기사용자들은 기정으로 보안조작체계의 __default__가입(이것은 보안조작체계사용자 user_u에 대응)에 대응되게 됩니다. 지령 useradd로 어떤 사용자가 만들어질 때 추가선택을 주지 않는 경우에는 보안조작체계사용자 user_u에 대응되게 됩니다. 다음의 실례는 기정대응관계를 정의합니다.

```
__default__                user_u                s0
```

제 2 절 useradd에 의한 새로운 봉사기사용자에 대한 제한 부여

《붉은별》봉사기용체계 3.0 판에서 useradd로 봉사기사용자들을 만들 때 추가선택 –Z를 사용하면 그 사용자가 어느 보안조작체계사용자와 대응되는가를 지적할수 있습니다. 다음의 실례에서는 useruuser라는 새로운 봉사기사용자를 만들고 그것을 보안조작체계사용자 staff_u에 대응시킵니다. 보안조작체계사용자 staff_u에 대응되는 봉사기사용자들은 령역 staff_t에서 실행됩니다. 이 령역에서 봉사기사용자들은 보안조작체계방책에서 허가하지 않는 한 setuid응용프로그람들을 실행할수 없으며(passwd와 같음) su나 sudo를 실행할수 없으며 결국 이러한 지령들을 리용하지 못하여 사용자들이 봉사기체계의 root사용자로 될수 없게 됩니다.

1. 체계관리자로서 지령 useradd -Z staff_u useruuser를 실행하여 보안조작체계사용자 staff_u에 대응되는 새로운 봉사기사용자(useruuser)를 만듭니다.

2. 보안관리자로서 지령 semanage login –l을 실행하여 봉사기사용자 useruuser와 staff_u사이의 대응관계를 볼수 있습니다.

```
# /usr/sbin/semanage  login  -l
```

```
가입사용자이름          보안조작체계사용자      MLS/MCS범위
__default__             user_u                  s0
root                    root                    s0-s15:c0.c1023
system_u                system_u                s0-s15:c0.c1023
useruuser               staff_u                 s0
```

체계관리자로서 지령 passwd useruuser를 실행시켜 봉사기사용자 useruuser에 통과암호를 할당합니다.

```
# passwd useruuser
useruuser사용자의 암호변경중
새 암호:
새 암호 다시입력:
passwd: 모든 인증통표가 성공적으로 갱신되였습니다.
```

3. 현재의 대화접속에서 탈퇴하고 봉사기체계의 useruuser라는 사용자로 가입합니다. 가입할 때 pam_selinux는 봉사기사용자를 보안조작체계 사용자로 대응(이 경우에는 staff_u)시키며 이때 생기게 되는 보안조작체계문맥을 설정합니다. 그다음 이 문맥을 가지고 봉사기사용자의 쉘이 실행되게 됩니다. 지령 id –Z를 실행시키면 봉사기사용자의 문맥을 볼수 있습니다.

```
[useruuser@localhost ~]$ id -Z
staff_u:staff_r:staff_t:s0
```

4. 봉사기체계의 useruuser의 대화접속에서 탈퇴하고 자기의 account로 다시 가입갑니다. 만일 봉사기체계의 사용자 useruuser를 삭제하려면 봉사기체계의 root사용자로 지령 userdel –r useruuser를 실행시키며 그의 사용자등록부도 삭제하려면 그 뒤에 사용자등록부를 쓰면 됩니다.

제 3 절 semanage login에 의한 현존사용자에 대한 제한 부여

만일 어떤 봉사기사용자가 보안조작체계사용자 user_u에 대응된다면 이때 대응되는 보안조작체계사용자를 변경시키기 위하여서는 지령 semanage login을 리용하여야 합니다. 아래의 실례에서는 newuser라는 새로운 봉사기사용자를 만들고 그것을 보안조작체계사용자 staff_u에 대응시킵니다.

1. 체계관리자자로서 지령 useradd newuser를 실행시켜 새로운 봉사기사

용자 newuser를 만듭니다. 이 사용자는 기정대응을 사용하기때문에 semanage login –l의 출력에 나타나지 않습니다.

```
#semanage login -l
가입사용자이름          보안조작체계사용자     MLS/MCS범위
__default__             user_u                 s0
root                    root                   s0-s15:c0.c1023
system_u                system_u               s0-s15:c0.c1023
```

2. 봉사기사용자 newuser를 보안조작체계사용자 staff_u에 대응시키려면 보안관리자로서 다음의 지령을 실행합니다.

```
#semanage login –a –s staff_u newuser
```

-a라는 추가선택은 새로운 기록을 추가한다는것이며 –s는 봉사기사용자에 대응하는 보안조작체계사용자를 지적합니다. 마지막인수인 newuser는 특정한 보안조작체계사용자에 대응시키려고 하는 봉사기사용자입니다.

3. 봉사기사용자 newuser와 staff_u사이의 대응관계를 보려면 보안관리자로 지령 semanage login –l을 실행합니다.

```
# /usr/sbin/semanage login -l
가입사용자이름          보안조작체계사용자     MLS/MCS범위
__default__             user_u                 s0
root                    root                   s0-s15:c0.c1023
system_u                system_u               s0-s15:c0.c1023
newuser                 staff_u                s0
```

체계관리자로 지령 passwd newuser를 실행하여 봉사기사용자 newuser에 통과암호를 할당합니다.

```
# passwd newuser
newuser사용자의 암호변경중
새 암호:
새 암호 다시입력:
passwd: 모든 인증통표가 성공적으로 갱신되였습니다.
```

4. 현재의 대화접속에서 탈퇴하고 봉사기사용자 newuser로 가입합니다. 지령 id –Z를 실행하여 newuser의 보안조작체계문맥을 볼수 있습니

다.

```
[newuser@localhost ~]$ id -Z
staff_u:staff_r:staff_t:s0
```

5. 봉사기체계의 newuser대화접속을 탈퇴하고 자신의 계산자리(account)로 도로 가입갑니다. 만일 봉사기체계의 사용자 newuser를 삭제하려면 체계관리자로 지령 userdel –r newuser를 실행시키며 그의 사용자등록부도 삭제하려면 그 뒤에 사용자등록부를 쓰면 됩니다. 또한 봉사기사용자 newuser와 user_u사이의 대응관계도 삭제됩니다.

```
# /usr/sbin/userdel -r newuser
# /usr/sbin/semanage login -l
가입사용자이름          보안조작체계사용자    MLS/MCS범위
__default__             user_u                s0
root                    root                  s0-s15:c0.c1023
system_u                system_u              s0-s15:c0.c1023
```

제 4 절 기정대응관계의 변경

《붉은별》봉사기용체계 3.0 판에서 봉사기사용자들은 기정으로 보안조작체계의 __default__가입(이것은 차례로 보안조작체계사용자 user_u에 대응)에 대응되게 됩니다. 만일 새로운 봉사기사용자들을 만들고 이때 어떤 보안조작체계사용자에 특별히 대응되지 않은 사용자들을 기정으로 보안조작체계사용자 staff_u에 대응시키려면 semanage login를 리용하여 기정대응관계를 변경시킬수 있습니다.

실례로 봉사기체계의 root사용자로서 다음의 지령들을 실행하면 기정대응관계를 user_u로부터 staff_u로 변경할수 있습니다.

```
#semanage login -m -S rss -s "staff_u" -r s0 __default__
```

보안관리자로서 지령 semanage login –l을 실행하여 __default__가입이 staff_u에 대응되는것을 확인할수 있습니다.

```
#semanage login -l
가입사용자이름        보안조작체계사용자    MLS/MCS범위
__default__           staff_u               s0
root                  root                  s0-s15:c0.c1023
system_u              system_u              s0-s15:c0.c1023
```

만일 새로운 봉사기사용자가 만들어지고 어떠한 보안조작체계사용자도 지적되지 않았다면 혹은 현존 봉사기사용자가 가입하고 그것이 semanage login –l의 출력물의 그 어떤 특정입구점과도 정합되지 않는다면 이러한 사용자들은 __default__가입에서처럼 staff_u에 대응되게 됩니다.

기정동작으로 도로 변경시키려면 봉사기체계의 root사용자로 다음의 지령을 사용하여 __default__가입을 보안조작체계사용자 user_u에 대응시켜야 합니다.

```
#semanage login -m -S rss -s "user_u" -r s0 __default__
```

제 5 절 사용자가 실행하는 응용프로그람에 대한 론리형

봉사기사용자들은 쓰기접근권한을 가지고있는 자기의 사용자등록부들과 /tmp/에서 응용프로그람(사용자의 권한을 계승하는 응용프로그람)들을 실행할수 없는데 이렇게 하면 결함이 있거나 악의적인 응용프로그람들이 사용자소유의 화일들을 변경하지 못하게 됩니다. 《붉은별》봉사기용체계 3.0판에서는 기정으로 령역 guest_t에 있는 봉사기사용자들이 자기의 사용자등록부들이나 /tmp/에서 응용프로그람들을 실행할수 없습니다. 그러나 령역 user_t와 staff_t에 있는 봉사기사용자들은 기정으로 자기의 사용자등록부들이나 /tmp/에서 응용프로그람들을 실행할수 있습니다.

론리형들은 이러한 동작을 변경시키는데 유효하며 setsebool지령으로 구성되게 됩니다. setsebool지령은 보안관리자로 실행되여야 합니다. 지령

setsebool –P는 영구적으로 변경시킵니다. 추가선택 –P를 사용하지 말아야 재기동되여도 변경이 보존되지 않습니다.

guest_t

령역 guest_t에 있는 봉사기사용자들이 자기의 사용자등록부들과 /tmp/에서 응용프로그람들을 실행할수 있게 하려면 다음과 같이 합니다.

```
#setsebool -P allow_guest_exec_content on
```

user_t

령역 user_t에 있는 봉사기사용자들이 자기의 사용자등록부들과 /tmp/에서 응용프로그람들을 실행하지 못하게 하려면 다음과 같이 합니다.

```
#setsebool -P allow_user_exec_content off
```

staff_t

령역 staff_t에 있는 봉사기사용자들이 자기의 사용자등록부들과 /tmp/에서 응용프로그람들을 실행하지 못하게 하려면 다음과 같이 합니다.

```
#setsebool -P allow_staff_exec_content off
```

제 6 장 프로쎄스에 대한 방책관리

응용프로그람에 대한 방책작성의 기본목적은 그 응용프로그람이 다른 응용프로그람이나 체계의 정상적인 기동에 영향을 주지 않도록 하자는데 있습니다.

1 절에서는 《붉은별》봉사기용체계 3.0 판에서 alternatives응용프로그람에 대한 보안방책작성을 통하여 응용프로그람에 대한 방책작성방법을 소개합니다.

2 절에서는 보안방책에서 제공하고있는 론리값을 리용한 봉사대몬의 접근권한관리방법을 소개합니다.

방책작성에서 반드시 지켜야 할 기본원칙은 《붉은별》봉사기용체계 3.0 판의 보안목표를 위반하지 않도록 하는것입니다.

제 1 절 응용프로그람에 대한 방책관리

방책모듈을 작성하기 전에 우선 응용프로그람에 대한 정보를 수집하고 검사환경을 설정하여야 합니다.

alternatives는 현재 기정보안방책에서 방책모듈로 정의되여있지 않다.

《붉은별》봉사기용체계 3.0 판에서는 다음과 같은 문제점을 해결하기 위하여 alternatives에 대한 방책모듈작성을 제안하게 됩니다.

1. 문제점

alternatives지령을 리용한 우편봉사기의 절환이 보안방책에 의해 허가되지 않습니다.

체계관리자(sysadm_r역할)는 다음과 같은 지령으로 전자우편봉사기를 절환하려고 합니다.

#alternatives –config mta

허가방식(permissive)에서 이 지령을 실행하면 다음과 같은 설정화면이 현시됩니다.

```
[root@localhost ~]# alternatives --config mta

2 개의 프로그람이 'mta'를 제공합니다.

  선택    명령
-----------------------------------------------
   1          /usr/sbin/sendmail.postfix
*+ 2          /usr/sbin/sendmail.sendmail
```

그림 6. alternativs프로그람의 실행

체계관리자는 수자 1 을 입력하여 우편봉사기를 postfix로 절환하려고 합니다.

만약 봉사기용체계가 시행방식(enforcing)으로 기동하고있는 경우에는 설정화면조차 현시되지 않습니다.

alternatives프로그람은 /usr/sbin등록부안의 기호련결화일 sendmail에 대한 속성값을 변경시킵니다. /usr/sbin/sendmail화일은 bin_t로 표식되여있으며 rss형보안방책에서는 체계에 대한 완전성보호를 실현하기 위하여 bin_t형에 대한 변경을 rpm_t형을 가진 패키지관리프로그람에만 허가하고있습니다.

2. alternatives프로그람의 접근권한

alternatives프로그람은 기정지령을 결정하기 위한 기호련결들을 유지관리하는 하나의 프로쎄스입니다.

- alternatives프로그람은 체계관리자에게 허가된 령역(sysadm_t)과 구별되는 독자적인 령역으로 동작하여야 합니다.
- alternatives프로그람은 체계관리자(sysadm_r역할)만이 실행할수 있습니다.
- alternatives프로그람은 /etc/alternatives등록부안의 기호련결화일들에 대한 쓰기권한을 요구합니다.
- alternatives프로그람은 /var/lib/alternatives등록부안에 있는 alternatives 상태정보화일들에 대한 읽기권한을 요구합니다.
- alternatives프로그람은 /usr/sbin등록부에 대한 쓰기권한을 요구합니다.
- alternatives프로그람은 /usr/lib등록부안의 화일들에 대한 쓰기권한을 요구합니다.
- alternatives프로그람은 /usr/share/man등록부안의 화일들에 대한 쓰기권한을 요구합니다.

3. 방책작성

rss형보안방책에서 매 방책모듈은 다음과 같은 3 개의 화일들로 이루어져있습니다.

형시행규칙화일(.te), 외부대면부화일(.if), 보안표식설정화일(.fc)

다음의 지령으로 다음과 같은 3 개의 화일을 /usr/share/selinux/devel등록부안에 생성합니다.

```
#cd /usr/share/selinux/devel
#touch alternatives.fc
#touch alternatives.if
#touch alternatives.te
```

다음과 같은 순서로 alternatives에 대한 방책을 작성합니다.

– 형선언

방책작성의 첫 단계는 방책모듈에 적합한 령역과 형을 선언하는것입니다.

alternatives방책모듈에서는 다음과 같은 형들을 선언합니다.

■ alternatives_t – alternatives응용프로그람(/usr/sbin/alternatives)에 대한 령역형

■ alternatives_exec_t – alternatives응용프로그람실행화일에 대한 형

■ alternatives_var_lib_t - /var/lib/alternatives등록부에 보관되는 화일들의 형

■ alternatives_etc_t - /etc/alternatives등록부에 보관되는 화일들의 형

alternatives_t와 alternatives_exec_t형을 제외한 나머지 형들은 alternatives응용으로그람에 의해 조종되는 자원들에 할당되는 형들입니다.

이상의 형들을 alternatives.te화일에서 다음과 같이 선언합니다.

매 형선언은 련관된 대면부를 호출합니다. 실례로 files_type라는 대면부는 alternatives_var_lib_t형으로 표식된 화일들이 일반화일속성을 가진다는것을 정의합니다.

```
policy_module(alternatives, 1.0)

########################################
# Declarations
type alternatives_t;
type alternatives_exec_t;
application_domain(alternatives_t, alternatives_exec_t)
role system_r types alternatives_t;

type alternatives_var_lib_t;
files_type(alternatives_var_lib_t)

type alternatives_etc_t;
files_config_file(alternatives_etc_t)
```

- 응용프로그람에 대한 초기 규칙작성

방책작성의 다음 단계는 alternatives령역을 위해 필요한 접근권한을 허가하는것입니다.

《2. alternatives프로그람의 접근권한》에 기초하여 다음과 같이 초기 규칙들을 추가합니다.

```
manage_dirs_pattern(alternatives_t, alternatives_var_lib_t, alternatives_var_lib_t)
manage_files_pattern(alternatives_t, alternatives_var_lib_t, alternatives_var_lib_t)

corecmd_exec_bin(alternatives_t)
corecmd_exec_shell(alternatives_t)
corecmd_manage_bin_files(alternatives_t)
corecmd_manage_all_executables(alternatives_t)

files_manage_etc_dirs(alternatives_t)
files_manage_etc_files(alternatives_t)
files_manage_etc_symlinks(alternatives_t)
```

```
init_read_all_script_files(alternatives_t)
init_getattr_all_script_files(alternatives_t)

kernel_dontaudit_read_system_state(alternatives_t)

libs_manage_lib_dirs(alternatives_t)
libs_manage_lib_files(alternatives_t)
libs_manage_lib_symlinks(alternatives_t)

miscfiles_manage_man_pages(alternatives_t)
miscfiles_read_localization(alternatives_t)
```

– 령역이행허가와 역할할당(Aloowing Domain Transition Authorizing Roles)

alternatives방책모듈에는 다음과 같은 령역이행규칙과 역할할당규칙이 포함됩니다.

■ sysadm_t령역이 alternatives_exec_t를 통하여 alternatives_t령역으로 이행하도록 허가하여야 합니다.

■ sysadm_r역할에 alternatives_t령역을 할당하여야 합니다.

alternatives.if화일에서 다음과 같은 대면부들을 정의합니다.

```
interface(`alternatives_domtrans',`
 gen_require(`
        type alternatives_t, alternatives_exec_t;
 ')
 domtrans_pattern($1, alternatives_exec_t, alternatives_t)
')
interface(`alternatives_run',`
 gen_require(`
        type alternatives_t;
 ')
 alternatives_domtrans($1)
 role $2 types alternatives_t;
')
```

이 대면부를 리용하여 sysadm.te화일에 다음과 같은 규칙을 추가합니다.

```
optional_policy(`
 alternatives_run(sysadm_t,  sysadm_r)
')
```

- 보안문맥할당규칙

alternatives.fc화일은 다음과 같이 작성합니다.

```
/usr/sbin/alternatives          --   gen_context(system_u:object_r:alternatives_exec_t,s0)
/usr/sbin/update-alternatives        -l   gen_context(system_u:object_r:alternatives_exec_t,s0)
/var/lib/alternatvies(/.*)?     --   gen_context(system_u:object_r:alternatives_var_lib_t,s0)
```

4. 설치와 시험

체계관리자로서 현재 작성된 방책을 콤파일합니다.

```
#make
```

이 지령에 의하여 /usr/share/selinux/devel등록부안에는 alternatives.pp화일이 생성됩니다.

생성된 alternatives.pp화일을 체계에 적재하고 fc화일에서 정의한 화일들에 대한 보안문맥을 변경합니다.

다음의 지령들은 보안관리자로서 실행하여야 합니다.

```
#semodule  –i  alternatives.pp
#restorecon  –F  –R  –v  /usr/sbin/alternatives
#restorecon  –F  –R  –v  /usr/sbin/update-alternatives
```

체계관리자로서 다음 지령이 성과적으로 실행되는가를 검사합니다.

```
#alternatives  –config  mta
```

주의:

현재 rss형보안방책에는 alternatives에 대한 방책모듈이 포함되여있습니다.

따라서 새로 작성된 alternatives방책모듈을 설치하는 경우 이미 작성되여있는 방책모듈을 갱신하게 됩니다.

제 2 절 론리형을 리용한 봉사대몬의 관리

rss형보안방책에서는 론리형부분품을 리용하여 allow규칙들을 동적으로 시행하거나 취소할수 있는 기능을 제공하고있습니다.

실례를 들어 samba봉사대몬에 대하여 다음과 같은 론리형부분품을 제공하고있습니다.

- allow_smbd_anon_write

  samba가 공개화일전송봉사를 위해 리용된 공개화일들을 변경하도록 허가합니다. 화일이나 등록부들은 public_content_rw_t로 표식되여있어야 합니다.

- samba_create_home_dirs

  samba가 새로운 홈등록부들(실례로 PAM을 리용)을 창조하도록 허가합니다.

- samba_domain_controller

  samba가 령역조종자로서 동작하며 사용자와 집단을 추가하고 암호를 변경하도록 허가합니다.

- samba_enable_home_dirs

  samba가 사용자홈등록부를 공유하도록 허가합니다.

- samba_export_all_ro

  samba가 임의의 화일이나 등록부를 읽기전용으로 공유하도록 허가합니다.

- samba_export_all_rw

  samba가 임의의 화일이나 등록부를 읽기 및 쓰기로 공유하도록 허가합니다.

- samba_run_unconfined

  samba가 제한을 받지 않는 스크립트들을 실행하도록 허가합니다.

- samba_share_nfs

  samba가 NFS volumes을 반출하도록 허가합니다.

- samba_share_fusefs

  samba가 ntfs/fusefs볼륨을 반출하도록 허가합니다.

제 7 장 조작시 제기되는 문제점

이 장에서는 보안조작체계가 접근을 거부할 때 어떤 일이 일어나는가를 설명합니다. 3가지 가장 중요한 문제점들인 정확한 표식할당에 대한 정보검색, 보안조작체계거부사건에 대한 분석, audit2allow로 전용방책모듈창조에 대한 문제점들을 론의합니다.

제 1 절 접근거부시 제기되는 문제점

접근을 허가하거나 거부하는것과 같은 보안조작체계판정은 완충기억됩니다. 이 완충기억기를 접근벡토르고속완충기(AVC)라고 합니다. 거부통보문들은 보안조작체계가 접근을 거부할 때 기록됩니다. 이와 같은 거부항목들은 《접근벡토르고속완충기 거부》라고도 하며 어느 대몬이 실행되고 있는가에 따라 여러가지 위치에 기록됩니다.

대몬	기록위치
auditd능동	/var/log/audit/audit.log
auditd 비능동, rsyslogd 능동	/var/log/messages
setroubleshootd, rsyslogd, auditd능동	/var/log/audit/audit.log. 간단한 거부통보문들은 /var/log/messages에 전송됩니다.

《붉은별》봉사기용체계 3.0판은 X체계를 지원하지 않습니다. 따라서 보안조작체계에 의해 접근이 거부될 때를 명확히 알수 없게 됩니다. 실례로

웨브싸이트를 열람하는 사용자들은 다음과 같은 오유를 받을수 있습니다.

```
Forbidden
You don't have permission to access file name on this server
```

이 상황에서 DAC규칙(표준리눅스권한)이 접근을 허가한다면 /var/log/messages와 /var/log/audit/audit.log에서 "SELinux is preventing"과 "denied"오유를 검색합니다. 이것은 보안관리자(secadm_r역할)로서 다음과 같은 지령을 실행하여 집행될수 있습니다.

```
#grep "SELinux is preventing" /var/log/messages
#grep "denied" /var/log/audit/audit.log
```

제 2 절 3가지 중요한 문제점

이 절에서는 3가지 중요한 문제점들인 표식할당문제, 봉사에 대한 론리값과 포구설정문제, 보안조작체계방책전개(evolving)문제를 설명합니다.

1. 표식할당문제

보안조작체계를 실행하고있는 체계들에서 모든 프로쎄스들과 화일들은 보안관련정보를 포함하고있는 표식을 받게 됩니다. 이 정보는 보안조작체계문맥이라고합니다. 만일 이 표식들이 정확하지 않다면 접근이 거부될수 있습니다. 또한 응용프로그람이 부정확하게 표식된다면 그것이 이행하는 프로쎄스 역시 정확한 표식을 가질수 없으며 보안조작체계는 접근을 거부하고 그 프로쎄스는 부정확하게 표식된 화일들을 창조할수 있습니다.

일반적으로 표식할당문제는 비표준등록부가 봉사에 리용되는 때에 제기됩니다. 실례로 관리자가 웨브싸이트에 대하여 /var/www/html을 리용하지 않고 /var/myweb/를 리용하려고 한다고 합시다. 《붉은별》봉사기용체계 3.0

관에서 /var/등록부는 var_t형으로 표식되여있습니다. 창조된 화일들과 등록부들, /myweb는 이 형을 계승합니다. 또한 새로 창조된 고준위등록부(/myserver/와 같은)들은 root_t형(restorecon지령에 의하여 default_t형)으로 표식될수 있습니다. 보안조작체계는 Apache HTTP봉사기가 이와 같은 형들에 접근하지 못하도록 보호합니다. 그와 같은 접근을 허가하기 위하여 보안조작체계는 /var/myweb/안의 화일들이 httpd에 접근될수 있는가를 알아야 합니다. 다음의 지령은 보안관리자로서 실행되여야 합니다.

```
#semanage fcontext -a -t httpd_sys_content_t "/var/myweb(/.*)?"
```

이 semanage지령은 /var/myweb/등록부에 대한 문맥을 보안조작체계문맥구성화일(file-context)에 추가합니다.[1] semanage지령은 문맥을 변경하지 않고 지령에서 입력한 값을 보안조작체계문맥구성화일안에 입구점으로 추가합니다. 체계관리자나 보안관리자로서 restorecon지령을 실행하여 그 변경내용을 적용합니다.

```
# /sbin/restorecon -R -v /srv/myweb
```

《제 4 장 보안조작체계구성과 관리》에서 《제 6 절 화일문맥설정》의 영《일관성변경》에서 file-context설정화일에 문맥을 추가하는 방법에 대하여 구체적으로 소개하고있습니다.

정확한 문맥은 무엇인가?

matchpathcon지령은 어떤 화일경로에 대한 문맥을 검사하고 그것을 그 경로에 대한 기정표식과 비교합니다. 다음의 실례에서는 부정

---

[1] /etc/selinux/targeted/contexts/files/안의 화일들은 화일들과 등록부들에 대한 문맥을 정의합니다. 이 등록부안의 화일들은 화일들과 등록부들을 기정문맥으로 회복하기 위하여 restorecon과 setfiles에 의해 읽어집니다.

확하게 표식된 화일들을 포함하고있는 등록부에 대한 matchpathcon 지령의 리용방법을 보여줍니다.

```
#matchpathcon  -V /var/www/html/*
/var/www/html/index.html has context root:object_r:admin_home_t:s0, shoul
d be system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html has context root:object_r:admin_home_t:s0, shou
ld be system_u:object_r:httpd_sys_content_t:s0
```

이 실례에서 index.html과 page1.html화일들은 admin_home_t화일로 표식됩니다. 이 형은 사용자등록부들안의 화일들을 위해 리용됩니다. mv지령을 리용하여 화일들을 사용자등록부로부터 이동하면 admin_home_t형으로 표식된 화일들을 생성할수 있습니다.. 이 형은 사용자등록부의 밖에서는 존재하지 않습니다. restorecon지령을 리용하여 그와 같은 화일들을 정확한 형으로 회복합니다.

```
#restorecon  -v /var/www/html/index.html
restorecon reset /var/www/html/index.html context root:object_r:admin_hom
e_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

등록부안의 모든 화일들에 대한 문맥회복은 –R선택항목을 리용하여 진행합니다.

```
#restorecon  -R -v /var/www/html/
restorecon reset /var/www/html/page1.html context root:object_r:samba_sha
re_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/www/html/index.html context root:object_r:samba_shar
e_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

《제 4 장 보안조작체계구성과 관리》에서 《제 8 절 보안조작체계표식보존》의 《3. 기정보안조작체계문맥을 검사》에서 matchpathcon지령에 대하여 구체적으로 설명하고있습니다.

2. 제한된 봉사는 어떻게 실행하고있는가?

봉사들은 다양한 방법으로 실행될수 있습니다. 이로부터 관리자는 봉사들을 실행하는 방법을 보안조작체계체계에 알려주어야 합니다. 이것은 보안조작체계방책작성에 대한 지식이 없이도 실행시에 보안조작체계방책의 일부가 변경되도록 허가하는 론리형부분품에 의해 실현될수 있습니다. 이와 같은 기능은 보안조작체계방책을 재적재하거나 재콤파일하지 않고도 NFS화일체계에 대한 봉사접근을 허가하는것과 같은 변경을 허가합니다. 또한 비기정포구번호들에서 실행하는 봉사대몬들은 semanage지령을 통하여 방책설정을 갱신할것을 요구합니다.

실례로 Apache HTTP봉사기가 MySQL과 통신하도록 허가하려면 httpd_can_network_connect_db론리값을 on으로 설정해야 합니다.

```
#setsebool -P httpd_can_network_connect_db on
```

만일 어떤 특별한 봉사에 대한 접근이 거부된다면 getsebool과 grep지령을 리용하여 어떤 론리값들이 접근을 허가할수 있는가를 알아보아야 합니다. 실례로 getsebool –a | grep ftp를 리용하여 FTP와 련관된 론리값들을 검색합니다.

```
#getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
```

론리형목록과 그것들의 상태(능동인가, 비능동인가)를 알아보려면 /usr/sbin/getsebool –a지령을 실행하면 됩니다. 론리값목록에 대하여 매 론리값에 대한 설명과 그것들의 상태를 알아보려면 보안관리자(secadm_r역할)로

서 semanage boolean –l지령을 실행하여야 합니다. 《제 4 장 보안조작체계 구성과 관리》의 《제 5 절 론리형값》에서 론리형목록과 설정에 대하여 구체적으로 서술하고있습니다.

- 포구번호들

방책설정(policy configuration)에 따라 봉사들이 일정한 포구번호들에서만 실행되도록 할수 있습니다. 방책을 변경하지 않고 봉사가 실행하는 포구를 변경하면 봉사를 기동할수 없게 됩니다. 실례로 semanage port –l | grep httpd지령을 보안관리자(secadm_r역할)로 실행하면 http와 련관된 포구들을 보여줍니다.

```
#semanage port -l | grep http
http_cache_port_t          tcp     3128, 8080, 8118
http_cache_port_t          udp     3130
http_port_t                tcp     80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t        tcp     5988
pegasus_https_port_t       tcp     5989
```

http_port_t포구형은 Apache HTTP봉사기가 청취할수 있는 포구들을 정의하며 여기서는 포구 80과 443, 488, 8008, 8009, 8443을 정의합니다. 만일 httpd가 포구 9876을 청취하도록 httpd.conf를 설정하고 방책에서 이것을 반영하지 않는다면 service httpd start지령은 실패하게 됩니다.

```
# /sbin/service httpd start
Starting httpd: (13)Permission denied: make_sock: could not bind to address [::]:9876
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:9876
no listening sockets available, shutting down
Unable to open logs                    [FAILED]
```

다음과 같은 보안조작체계거부사건이 /var/log/audit/audit.log화일에 기록됩니다.

```
type=AVC msg=audit(1225948455.061:294): avc: denied { name_bind } for pid=4997 comm="httpd" src=9876 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

httpd가 http_port_t포구형으로 설정되지 않은 포구에서 청취하도록 허가하려면 semanage port지령을 실행하여 방책구성화일에 포구를 추가해야 합니다.[2] 이 지령은 보안관리자로서 실행하여야 합니다.

```
#semanage port -a -t http_port_t -p tcp 9876
```

-a 선택항목은 새로운 레코드를 추가하며 –t선택항목은 형을 정의합니다. –p 선택항목은 포구를 정의합니다. 마지막 인수는 추가하려는 포구번호입니다.

3. 규칙갱신과 파괴된 응용프로그람들

응용프로그람이 파괴될수도 있는데 이때 보안조작체계는 그 응용프로그람의 접근요구를 거부할수 있습니다. 또한 보안조작체계방책들이 갱신(evolve)되면 보안조작체계는 일정한 방식으로 실행하고있는 응용프로그람을 인식하지 못할수 있으며 결과 응용프로그람이 기대한대로 동작한다고 해도 접근거부를 일으킬수 있습니다. 실례로 새로운 판본의 PostgreSQL이 배포된다면 그것은 현재 방책이 이전에는 본적이 없는 동작들을 수행할수 있으며 결과 접근이 거부될수 있습니다.

이와 같은 상황에서 접근이 거부된 다음에는 audit2allow지령을 리용하여 접근을 허가하는 전용방책모듈을 창조합니다. 이장의 《제 3 절 문제점

[2] semanage port –a지령은 /etc/selinux/targeted/modules/active/ports.local화일에 입구점을 추가합니다.

주의: 기정으로 이 화일은 보안관리자(secadm_r역할)에 의해서만 볼수 있습니다.

수정》의 《8. 접근허가:audit2allow》에서 audit2allowa지령에 대하여 구체적으로 설명하고있습니다.

제 3 절 문제점수정(Fixing Problems)

여기서는 오유수정방법들을 취급합니다. 보안조작체계규칙에 앞서 진행되는 Linux권한검사, 기록되지 않는 접근거부사건에 대한 원인분석, 표식할당과 론리값에 대한 정보를 포함하고있는 봉사에 대한 사용설명서, 전반적인 체계가 아니라 하나의 프로쎄스만을 허가방식으로 실행하도록 하는 허가방식의 령역, 거부사건들에 대한 분석, audit2allow를 리용한 전용방책모듈의 작성등을 론의합니다.

1. Linux권한

접근이 거부될 때 표준 Linux권한을 검사합니다. 《제 1 장 소개》에서 언급한것과 같이 대부분의 조작체계들은 자유접근조종(DAC)을 리용하여 접근을 조종하며 사용자가 자기가 소유하고있는 화일들의 권한을 조종하도록 허가합니다. 보안조작체계규칙들은 자유접근조종규칙들 다음에 검사됩니다. 보안조작체계규칙들은 DAC규칙들이 접근을 거부하면 리용되지 않습니다.

만일 접근이 거부되고 보안조작체계거부사건들이 기록되지 않는 다면 ls –l지령을 리용하여 표준 Linux권한을 조사하여야 합니다.

```
#ls -l /var/www/html/index.html
-rw-r----- 1 root root 0 2009-05-07 11:06 index.html
```

이 실례에서 index.html은 root사용자와 집단에 속하고있습니다. root사용자는 읽기와 쓰기권한(-rw)을 가지며 root집단의 성원들은 읽기권한(-r-)을

가지고있습니다. 그밖의 다른 성원들은 권한(---)을 가지지 않습니다. 기정으로 이와 같은 권한들은 httpd가 이 화일을 읽지못하도록 합니다. 이 문제점을 해결하려면 chown지령을 리용하여 소유자와 집단을 변경시켜야 합니다. 이 지령은 체계관리자(sysadm_r역할)로서 실행되여야 합니다.

```
# chown apache:apache /var/www/html/index.html
```

이것은 기정설정값을 취하는것이며 여기서 httpd는 Linux Apache사용자로서 실행하도록 되여있습니다. 만일 httpd를 다른 사용자로 실행하려고 한다면 apache:apache를 그 사용자로 교체하면 됩니다.

2. silent거부사건(Possible Causes of Silent Denials)들에 대한 원인분석

어떤 상황에서 접근벡토르고속기억완충기(AVC)거부사건들은 보안조작체계가 접근을 거부할 때 기록되지 않을수도 있습니다. 응용프로그람들과 체계서고함수들은 자주 자기의 과제를 수행하는데 필요한것보다 더 많은 접근을 요구합니다. 해롭지 않은 응용프로그람조사기능에 대한 AVC거부사건들로 audit기록화일을 채우지 않고 최소특권을 유지하기 위하여 방책은 dontaudit규칙들을 리용하여 권한을 허가하지 않고 AVC거부사건들이 기록되지 않도록 할수 있습니다. Dontaudit의 결함은 보안조작체계가 접근을 거부한다고 해도 거부통보문들이 기록되지 않으며 고장퇴치를 어렵게 한다는것입니다.

림시적으로 dontaudit규칙들을 사용불가능하게 설정하고 모든 거부사건들을 기록시키려고 한다면 보안관리자(secadm_r역할)로 다음과 같은 지령을 실행하여야 합니다.

```
#semodule –DB
```

-D선택항목은 dontaudit규칙들을 사용할수 없게 하며 –B선택항목은 방책을 재구축합니다. semodule –DB지령을 실행한 다음 접근권한문제점을 발생시킨 응용프로그람을 다시 실행하여 그 응용프로그람과 련관된 보안조작체계거부사건들이 기록되는가를 알아봅니다. 일부 거부사건들이 dontaudit규칙들을 통하여 무시되고 조종되기때문에 어느 거부사건들이 허가되여야 하는가를 결정하는데 주의를 돌려야 합니다.

방책을 재구축하고 dontaudit규칙들을 사용가능하게 설정하려면 보안관리자(secadm_r역할)로 다음의 지령을 실행하여야 합니다.

```
#semodule –B
```

이것은 방책을 초기상태로 회복합니다. dontaudit규칙에 대한 충분한 목록은 sesearch –dontaudit지령을 실행하여 알아볼수 있습니다. –s domain선택항목과 grep지령을 리용하여 검색범위를 좁힐수 있습니다.

```
#sesearch --dontaudit -s smbd_t | grep squid
WARNING: This policy contained disabled aliases; they have been removed.
dontaudit smbd_t squid_port_t : tcp_socket name_bind ;
dontaudit smbd_t squid_port_t : udp_socket name_bind ;
```

(이 지령은 setools패키지를 설치하여야 동작합니다.)

《6. 가공하지 않은 audit통보문》과 《7. sealert통보문》을 참고하여 거부사건분석에 대한 정보를 알아볼수 있습니다.

3. 봉사에 대한 사용설명서

봉사에 대한 사용설명서는 주어진 상황에서 어떤 화일형을 리용하는가와 같은 가치있는 정보와 봉사가 가지고있는 접근변경 론리값을 포함하고 있습니다.(NFS화일체계에 접근하는 httpd와 같은것) 이 정보는 표준 안내페지나 보안조작체계에 대한 안내페지에 있을수 있습니다.

실례로 httpd_selinux(8)안내페지는 주어진 상황에서 어떤 화일형을 리용하는가에 대한 정보뿐만아니라 스크립트, 화일공유, 사용자등록부안의 등록부접근을 허가하는 론리값, 등에 대한 정보들을 포함하고있습니다. 봉사대몬들에 대하여 보안조작체계정보를 포함하고있는 기타 안내페지들은 다음과 같습니다.

- Samba: samba_selinux(8)안내페지는 Samba를 통하여 반출되는 화일과 등록부들이 samba_share_t형으로 표식되여야 한다는것을 설명할 뿐만아니라 samba_share_t가 아닌 다른 형으로 표식된 화일들이 Samba를 통하여 반출되도록 하는 론리값에 대한 설명도 제시합니다.
- NFS: nfs_selinux(8)안내페지는 기정으로 화일체계들이 NFS를 통하여 반출될수 없다는것과 화일체계들이 반출되도록 허가하기 위하여 nfs_export_all_ro나 nfs_export_all_rw와 같은 론리값이 능동으로 되여야 한다는것을 서술하고있습니다.
- Berkeley Internet Name Domain(BIND): named(8)안내페지는 주어진 상황에서 어떤 화일형이 리용되는가를 서술합니다. named_selinux(8)안내페지는 기정으로 named는 주령역화일(master zone)들에 쓰기할수 없으며 그와 같은 접근을 허가하려면 named_write_master_zones론리값이 능동으로 되여야 한다는것을 서술하고있습니다.

안내페지안의 정보들은 정확한 화일형과 론리값들을 설정하도록 하며 보안조작체계가 접근을 거부하지 못하도록 합니다.

4. 허가방식의 령역

보안조작체계가 허가방식으로 동작할 때 보안조작체계는 접근을 거부하지 않지만 시행방식으로 동작하는 경우에 거부되는 동작들에 대한 사건이 기록됩니다. 이전에는 하나의 령역을 허가방식으로 설정할수 없었다. 이

것은 어떤 상황에서는 문제점을 수정하기 위하여 체계전반을 허가방식으로 설정하여야 하였습니다.

《붉은별》봉사기용체계 3.0 판은 허가방식의 령역들을 포함하고있으며 거기서는 관리자가 체계전반을 허가방식으로 설정하는것이 아니라 하나의 프로쎄스가 허가방식으로 실행하도록 설정할수 있습니다. 보안조작체계검사는 허가방식의 령역에 대하여서도 여전히 진행됩니다. 그러나 핵심부는 접근을 허가하고 보안조작체계가 접근을 거부한 상황에 대하여서는 AVC거부사건을 보고합니다.

domain_disable_trans론리값은 응용프로그람이 제한을 받는 령역으로 이행하지 못하도록 하는데 리용할수 있습니다. 따라서 그 프로쎄스는 initrc_t와 같은 제한을 받지 않는 령역으로 실행합니다. 그와 같은 론리값을 능동으로 설정하는것은 중요한 문제점을 일으킬수 있습니다. 실례로 httpd_disable_trans론리값이 능동으로 된다면

- httpd는 제한을 받지 않는 initrc_t령역으로 실행합니다. initrc_t령역으로 실행하고있는 프로쎄스들에 의해 창조된 화일들은 httpd_t령역으로 실행하고있는 프로쎄스에 의해 창조된 화일과 같은 표식할당규칙들을 가질수 없으며 이것은 프로쎄스들이 부정확하게 표식된 화일들을 창조하도록 할수 있습니다. 이것은 이후에 접근문제점을 발생시킵니다.
- httpd_t와 통신하도록 허가된 제한을 받는 령역들은 initrc_t와 통신할수 없으며 실패할수 있습니다.

허가방식의 령역들은 다음과 같은 경우에 리용될수 있습니다.

- 어떤 문제점을 퇴치하기 위하여 전반적인 체계를 허가방식으로 설정하여 체계전반을 위험상태로 만드는것이 아니라 하나의 프로쎄스

(령역)를 허가방식으로 실행하도록 설정하는 경우

● 새로운 응용프로그람들에 대한 방책을 창조하는 경우. 이전에는 최소 방책이 창조되면 체계전반이 허가방식으로 되여 응용프로그람이 실행되고 SELinux거부사건들이 기록되도록 하였습니다. 그 다음 audit 2allow가 방책작성에 리용될수 있습니다. 이것은 체계전반을 위험에 빠지게 합니다. 허가방식의 령역으로는 새로운 방책안에 있는 령역만이 허가방식으로 표식될수 있도록 하며 체계전반은 위험상태에 놓이지 않게 됩니다.

- 령역을 허가방식으로 설정

령역을 허가방식으로 설정하려면 semanage permissive –a domain지령을 실행하여야 하며 여기서 domain은 허가방식으로 설정하려고 하는 령역입니다. 실례로 다음과 같은 지령을 Linux root사용자로 실행하면 httpd_t령역(Apache HTTP봉사기가 실행하는 령역)을 허가방식으로 설정합니다.

```
#semanage permissive –a httpd_t
```

허가방식으로 설정한 령역목록을 보려면 semodule –l | grep permissive지령을 보안관리자로 실행하여야 합니다. 실례로

```
#semodule –l | grep permissive
perimissive_httpd_t 1.0
```

만일 어떤 령역이 허가방식으로 되는것을 더이상 요구하지 않는다면 semanage permissive –d domain지령을 보안관리자로 실행하여야 합니다. 실례로

```
#semanage permissive –d httpd_t
```

- 허가방식령역에 대한 거부사건

SYSCALL통보문은 허가방식의 령역에만 대응하는 통보문입니다. 다음과 같은 실례는 Apache HTTP봉사기에서의 AVC거부사건(그리고 련관된 체계호출)을 보여줍니다.

```
type=AVC msg=audit(1226882736.442:86): avc: denied { getattr } for
 pid=2427 comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=2841
33 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object
_r:samba_share_t:s0 tclass=file

type=SYSCALL msg=audit(1226882736.442:86): arch=40000003 syscall=19
6 success=no exit=-13 a0=b9a1e198 a1=bfc2921c a2=54dff4 a3=2008171
items=0 ppid=2425 pid=2427 auid=502 uid=48 gid=48 euid=48 suid=48 f
suid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/
usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

기정으로 httpd_t령역은 허가방식이 아니며 따라서 동작이 거부되게 되며 SYSCALL통보문은 success=no를 포함합니다. 다음의 실례는 semange permissive –a httpd지령이 httpd_t령역을 허가방식으로 설정하도록 실행되였다는것을 제외하고는 같은 상황에 대한 AVC거부사건입니다.

```
type=AVC msg=audit(1226882925.714:136): avc: denied { read } for
pid=2512 comm="httpd" name="file1" dev=dm-0 ino=284133 scontext=unc
onfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t
:s0 tclass=file

type=SYSCALL msg=audit(1226882925.714:136): arch=40000003 syscall=5
 success=yes exit=11 a0=b962a1e8 a1=8000 a2=0 a3=8000 items=0 ppid=
2511 pid=2512 auid=502 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=
48 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

이 경우에 비록 AVC거부사건이 기록되였다고 해도 SYSCALL통보문에서 success=yes로 보여준것과 같이 접근은 거부되지 않았다.

허가방식의 령역에 대한 정보는 Dan Walsh의 "Permissive Domains"를 참고하면 됩니다.

5. 거부사건의 검색과 보기

ausearch와 aureport, sealert와 같은 많은 도구들이 보안조작체계거부사건들을 검색하고 보는데 리용될수 있습니다.

- ausearch

audit패키지는 ausearch지령을 제공합니다. ausearch(8)안내페지에서는 《ausearch는 여러가지 검색기준에 기초하여 사건들에 대한 감시 및 기록 대몬자료들을 문의할수 있는 도구입니다.》라고 서술되여있습니다. ausearch도구는 /var/log/audit/audit.log에 접근하며 따라서 Linux root사용자로서 실행되여야 합니다.

검색항목	지령
모든 거부사건들	/sbin/ausearch -m avc
금일에 대한 거부사건들	/sbin/ausearch -m avc -ts today
지난 10분동안의 거부사건들	/sbin/ausearch -m avc -ts recent

특별한 봉사에 대하여 보안조작체계거부사건들을 검색하려면 –c comm.-name선택항목을 리용하며 여기서 comm.-name은 실행화일의 이름입니다. 실례로 Apache HTTP봉사기에 대하여서는 httpd, Samba에 대하여서는 smbd입니다.

```
#ausearch -m avc -c httpd
#ausearch -m avc -c smbd
```

ausearch(8)안내페지를 참고하면 ausearch선택항목에 대하여 더 잘 알수 있습니다.

- aureport

audit패키지는 aureport를 제공합니다. aureport(8)안내페지에서는 《aurepor t가 audit체계기록사건들에 대한 상세한 보고서를 작성하는 도구입니다.》 라고 되여있습니다. aureport도구는 /var/log/audit/audit.log화일에 접근하며 따라서 Linux root사용자로 실행되여야 합니다. 보안조작체계거부목록을 보고 매 사건이 얼마나 자주 발생하였는가를 보려면 aureport –a지령을 실행하여야 합니다. 다음의 실례는 두가지 거부사건들을 보여줍니다.

```
#aureport -a
AVC Report
========================================================
# date time comm subj syscall class permission obj event
========================================================
1. 05/01/2009 21:41:39 httpd unconfined_u:system_r:httpd_t:s0 195 file ge
tattr
 system_u:object_r:samba_share_t:s0 denied 2
2. 05/03/2009 22:00:25 vsftpd unconfined_u:system_r:ftpd_t:s0 5 file read
 unconfined_u:object_r:cifs_t:s0 denied 4
```

aureport(8)안내페지를 참고하면 aureport선택항목에 대하여 더 잘 알수 있습니다.

6. 가공하지 않은(raw) Audit통보문

가공하지 않은 audit통보문들은 /var/log/audit/audit.log에 기록됩니다. 다음의 실례는 Apach HTTP봉사기(httpd_t령역으로 실행)가 /var/www/html/file1화일(samba_share_t형으로 표식)에 접근하려고 시도하였을 때 발생한 AVC 거부사건(체계호출과 련관된)을 보여줍니다.

```
type=AVC msg=audit(1226874073.147:96): avc: denied { getattr } for
 pid=2465 comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=2841
33 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object
_r:samba_share_t:s0 tclass=file
type=SYSCALL msg=audit(1226874073.147:96): arch=40000003 syscall=19
6 success=no exit=-13 a0=b98df198 a1=bfec85dc a2=54dff4 a3=2008171 i
tems=0 ppid=2463 pid=2465 auid=502 uid=48 gid=48 euid=48 suid=48 fs
```

```
uid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=6 comm="httpd" exe="/u
sr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

{ getattr }

괄호안의 이 항목은 거부된 권한을 나타냅니다. getattr는 원천프로쎄스가 대상화일의 상태정보를 읽으려고 시도하였다는것을 나타냅니다. 이것은 화일을 읽기전에 발생합니다. 이 동작은 부정확한 표식을 가지는 화일에 접근하는것으로 하여 거부되게 됩니다. 공통적으로 보게되는 권한들은 getattr와 read, write권한들입니다.

comm="httpd"

프로쎄스를 기동한 실행화일입니다. 실행화일의 완전경로는 체계호출(SYSCALL)통보문의 exe=부분에서 찾아볼수 있으며 여기서는 exe="/usr/sbin/httpd"입니다.

path="/var/www/html/file1"

프로쎄스가 접근하려고 하는 객체(대상)의 경로입니다.

scontext="unconfined_u:system_r:httpd_t:s0"

거부된 동작을 시도한 프로쎄스의 보안조작체계문맥입니다. 이 경우에는 Apache HTTP봉사기의 보안조작체계문맥이며 그것은 httpd_t령역으로 실행하고있습니다.

tcontext="unconfined_u:object_r:samba_share_t:s0"

프로쎄스가 접근을 시도한 객체(대상)의 보안조작체계문맥입니다. 이 경우에는 file1 의 보안조작체계문맥입니다. 주의해야 할것은 samba_share_t형은 httpd_t령역으로 실행하고있는 프로쎄스들이 접근할수 없다는것입니다.

어떤 상황에서는 tcontext가 scontext와 일치할수 있습니다. 실례로 프로쎄스가 어떤 체계봉사를 실행하려고 할 때입니다. 이때 그 체계봉사는 프로쎄스를 실행하는 봉사의 특징(실례:사용자식별자)을 변경하려고 합니다. 또한 tcontext는 다음과 같은 경우에 scontext와 일치할수 있습니다. 프로쎄스가 일반적인 제한보다 더 많은 자원(기억기와 같은)을 리용하려고 할 때입니다. 이것은 프로쎄스가 그 제한을 파괴하도록 허가되는가를 알아보는 보안검사로 됩니다.

체계호출(SYSCALL)통보문에서 중요한것은 다음의 2가지 항목들입니다.

- success=no: 거부사건(AVC)이 시행되였는가 시행되지 않았는가를 나타낸다. success=no는 체계호출이 성공하지 못하였다는것을 나타낸다.(보안조작체계가 접근을 거부하였다) success=yes는 체계호출이 성공하였다는것을 나타낸다. 이것은 허가방식의 령역이나 initrc_t와 kernel_t와 같은 허가방식의 령역에서만 볼수 있습니다.
- exe="/usr/sbin/httpd": 프로쎄스를 기동한 실행화일의 완전경로이며 이 실례에서는 exe="/usr/sbin/httpd"입니다.

부정확한 화일형은 보안조작체계가 접근을 거부하는 기본원인으로 됩니다. 고장퇴치를 진행하려면 원천문맥(scontext)과 대상문맥(tcontext)을 비교하여야 합니다. 그 프로쎄스(scontext)가 그와 같은 객체(tcontext)에 접근할수 있는가? 실례로 Apache HTTP봉사기(httpd_t)는 다른 방법으로 설정되지 않는 한 httpd_sys_content_t와 public_content_t, 등과 같이 httpd_selinux(8)안내페지에서 정의한 형들에만 접근할수 있습니다.

7. 접근허가: audit2allow

제품제작시에는 이 절에서 제시한 실례를 리용하지 말아야 합니다. 여기서는 오직 audit2allow지령의 리용방법만을 보여줍니다.

audit2allow(1)안내페지에서는 《audit2allow는 거부된 조작의 기록화일로부터 보안조작체계방책 허가규칙들을 생성합니다.》라고 서술되여있습니다. 거부사건들을 분석한 다음 표식변경을 진행하지 않거나 론리값에 의한 접근허가를 진행하지 않은 경우 audit2allow지령을 리용하여 국부방책모듈을 창조합니다. 접근이 보안조작체계에 의해 거부된 다음 audit2allow지령을 실행하면 이전에 거부된 접근을 허가하는 형시행규칙들을 보여줍니다.

다음의 실례에서는 audit2allow지령을 리용하여 방책모듈을 창조하는 실례를 보여줍니다.

1. 거부사건과 련관된 체계호출은 /var/log/audit/audit.log에 기록됩니다.

```
type=AVC msg=audit(1226270358.848:238): avc: denied { write } for pid=13349 comm="certwatch" name="cache" dev=dm-0 ino=218171 scontext=system_u:system_r:certwatch_t:s0 tcontext=system_u:object_r:var_t:s0 tclass=dir
type=SYSCALL msg=audit(1226270358.848:238): arch=40000003 syscall=39 success=no exit=-13 a0=39a2bf a1=3ff a2=3a0354 a3=94703c8 items=0 ppid=13344 pid=13349 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="certwatch" exe="/usr/bin/certwatch" subj=system_u:system_r:certwatch_t:s0 key=(null)
```

이 실례에서 certwatch(comm="certwatch")는 var_t형(tcontext=system_u:object_r:var_t:s0)으로 표식된 등록부에 대한 쓰기접근이 거부되였습니다. 거부사건을 분석합니다. 표식변경이나 론리값으로의 접근허가가 되지 않는 경우 audit2allow지령으로 국부방책모듈을 창조합니다.

2. 단계 1 에서 certwatch거부사건과 같은 기록된 거부사건을 가지고 audit2allow –w –a지령을 실행하면 접근이 거부된 원인을 리해하기 쉬운 서술형식으로 보여줍니다. –a선택항목은 모든 audit기록화일을 읽도록 하며 –w선택항목은 리해하기 쉬운 서술형식으로 생성하도록 합니다. audit2allow지령은 /var/log/audit/audit.log에 접근하며 따라서 체

계관리자로 실행되여야 합니다.

```
# audit2allow  -w -a
type=AVC msg=audit(1226270358.848:238): avc:  denied  { write } for
 pid=13349 comm="certwatch" name="cache" dev=dm-0 ino=218171 scont
ext=system_u:system_r:certwatch_t:s0 tcontext=system_u:object_r:var_t:s0 tc
lass=dir
 Was caused by:
 Missing type enforcement (TE) allow rule.
 You can use audit2allow to generate a loadable module to allow this ac
cess.
```

이상과 같이 형시행규칙이 없기때문에 접근이 거부되였습니다.

3. audit2allow –a지령을 리용하여 거부된 접근을 허가하는 형시행규칙을 볼수 있습니다.

```
# audit2allow  -a
#============= certwatch_t ==============
allow certwatch_t var_t:dir write;
```

4. audit2allow –a에 의해 현시되는 규칙들을 리용하려면 audit2allow –a –M mycertwatch지령을 Linux root사용자로서 실행하여야 하며 이 지령은 전용모듈을 창조합니다. –M선택항목은 현재 작업등록부안에 –M으로 정의한 이름으로 형시행규칙화일(.te)을 창조합니다.

```
# audit2allow  -a -M mycertwatch
******************** IMPORTANT ***********************
To make this policy package active, execute:
semodule  -i mycertwatch.pp
# ls
mycertwatch.pp   mycertwatch.te
```

또한 audit2allow는 형시행규칙을 방책패키지(.pp)로 콤파일합니다. 모듈을 설치하기 위하여 /usr/sbin/semodule –i mycertwatch.pp지령을 Linux root사용자로서 실행합니다.

만일 여러 프로쎄스들로 인한 여러가지 거부사건들이 존재하지만 하나의 프로쎄스에 대한 전용방책을 창조하려고 한다면 grep지령을 리

용하여 audit2allow에 대한 입력범위를 좁힐수 있습니다. 다음의 실례에서는 audit2allow를 통하여 certwatch와 련관된 거부사건들만을 전송하기 위하여 grep를 리용하는 실례를 보여줍니다.

```
# grep certwatch /var/log/audit/audit.log | audit2allow -M mycertwatch2
********************* IMPORTANT ************************
To make this policy package active, execute:
# /usr/sbin/semodule -i mycertwatch2.pp
```

audit2allow리용방법에 대한 보다 상세한 정보를 서술한 책들이 있으므로 그것을 참고하면 더욱 좋습니다.